



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - K141502

RANCANG BANGUN APLIKASI PANDUAN WISATA MAGELANG DI ERA MILLENNIAL BERBASIS ANDROID MOBILE APP

NURUL AZIZAH
NRP 5114100090

Dosen Pembimbing
Fajar Baskoro, S.Kom., MT.
Adhatas Solichah Ahmadiyah, S.Kom, M.Sc

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - K141502

RANCANG BANGUN APLIKASI PANDUAN WISATA MAGELANG DI ERA MILLENNIAL BERBASIS ANDROID MOBILE APP

NURUL AZIZAH
NRP 5114100090

Dosen Pembimbing
Fajar Baskoro, S.Kom, MT.
Adhatus Solichah Ahmadiyah, S.Kom, M.Sc

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - K141502

DESIGN AND IMPLEMENTATION OF MAGELANG TOURISM GUIDE IN MILLENNIAL ERA BASED ON ANDROID MOBILE APP

NURUL AZIZAH
NRP 5114100090

Dosen Pembimbing
Fajar Baskoro, S.Kom, MT.
Adhatus Solichah Ahmadiyah, S.Kom, M.Sc

DEPARTMENT OF INFORMATICS
Faculty of Information and Communication Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI PANDUAN WISATA MAGELANG DI ERA MILLENNIAL BERBASIS ANDROID MOBILE APP

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

NURUL AZIZAH
NRP : 5114 100 090

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Fajar Baskoro, S.Kom., M.T.

NIP: 19740403 199903 1 002

(pembimbing 1)

Adhatus Solichah A, S.Kom., Sc.

NIP: 19850826 201504 2 002

(pembimbing 2)

SURABAYA
JUNI 2018

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN APLIKASI PANDUAN WISATA MAGELANG DI ERA MILLENNIAL BERBASIS ANDROID MOBILE APP

Nama Mahasiswa : Nurul Azizah
NRP : 5114 100 090
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing 1 : Fajar Baskoro, S.Kom, MT.
Dosen Pembimbing 2 : Adhatus Solichah Ahmadiyah, S.Kom, M.Sc

ABSTRAK

Magelang merupakan daerah yang mempunyai potensi wisata yang sangat tinggi terhitung lebih dari 80 destinasi wisata terdapat pada daerah ini, Hal ini menjadikan Magelang salah satu tujuan wisata pilihan di Indonesia bahkan Mancanegara, akan tetapi para wisatawan dirasa masih kesulitan dalam memilih destinasi wisata yang cocok dan sesuai keinginannya selain itu informasi up to date terkait destinasi wisata pun juga masih sulit didapat, terlebih di jaman era millennial dimana penggunanya hobi mengeksplor tempat-tempat wisata baru dan menyukai sesuatu yang praktis.

Untuk mengatasi masalah tersebut, pada tugas akhir ini dibangun sebuah sistem yang terdiri dari aplikasi perangkat bergerak berbasis Android yang nantinya digunakan untuk pengguna dan aplikasi berbasis web yang digunakan untuk administrator. Aplikasi ini nantinya dapat mengelompokkan tempat wisata yang ada di Magelang berdasarkan kebutuhan pengguna era millennial. Tujuan kedua yaitu menyajikan informasi lengkap dan up to date terkait wisata yang dipilih, selain itu pengguna juga dapat melihat dan memberikan ulasan terkait destinasi wisata yang dipilih, tak hanya itu, pengguna juga dapat memberikan rekomendasi wisata baru ataupun update info terbaru

terkait wisata yang sudah ada kepada administrator serta pengguna juga dapat mengunggah foto spot wisata pada galeri, sedangkan pada sisi administrator, administrator berfungsi untuk mengelola konten wisata, melakukan manajemen pengguna, menanggapi rekomendasi wisata dari pengguna dan yang terakhir sebagai kontroler terhadap komentar negatif.

Dengan adanya sistem ini, diharapkan wisatawan dapat dengan mudah menemukan destinasi wisata yang sesuai dengan kriteria yang diinginkan sesuai kebutuhan pengguna era millennial, begitupula Dinas Pariwisata Magelang yang akan terbantu dalam promosi wisata Magelang dan juga dalam memperbaiki maupun meningkatkan kualitas wisata dari hasil review pengunjung pada Aplikasi ini nantinya.

Kata kunci: Pariwisata, Magelang, Android, Web, Millennial.

DESIGN AND IMPLEMENTATION OF MAGELANG TOURISM GUIDE IN MILLENNIAL ERA BASED ON ANDROID MOBILE APP

Student Name : Nurul Azizah
Student ID : 5114 100 090
Major : Informatics Department FTIf-ITS
Advisor 1 : Fajar Baskoro, S.Kom, MT.
Advisor 2 : Adhatus Solichah Ahmadiyah, S.Kom, M.Sc

ABSTRACT

Magelang is a region that has a very high tourism potential of more than 80 tourist destinations located in this region. This makes Magelang one of the preferred tourist destinations in Indonesia even abroad. However the tourists are still difficult in choosing the appropriate tourist destinations according to their desire, also up to date information related to tourist destinations is still difficult to obtain. Furthermore, In this millennial era who love to explore new tourist attractions and love something practical.

To solve the problems, in this final project built a system consisting of Android based mobile device application that will be used for users and web based applications used for administrators. This application will later be able to classify the place of tourist destinations in Magelang based on the current trend of kids on millennial era. The second objection is to provide up to date informations related to selected tours. In addition users can also view and give reviews related to selected tourist destinations. Not only that, users can also give new tours recommendations or update information about tours to the administrator also users can upload their tours photo to the gallery. While on the administrator side, administrator serves to manage travel content, perform user management, respond to user recommendations and control negative comments.

With this system, it is expected that tourists can easily find tourist destinations according to the criteria they want based on current trend millennial era. As well as the Magelang Tourism Office that will be helpful in the promotion of Magelang Tourism and also in fixing and improving the quality of tours from the results of visitor reviews on this application later.

Keywords: Tourism, Magelang, Android, Web, Millennial.

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan atas kehadiran Allah SWT, yang oleh karena anugrahNya penulis dapat menyelesaikan tugas akhir yang berjudul “RANCANG BANGUN APLIKASI PANDUAN WISATA MAGELANG DI ERA MILLENNIAL BERBASIS ANDROID MOBILE APP”.

Pengerjaan tugas akhir ini menjadi suatu pengalaman yang baik bagi penulis. Penulis dapat memperoleh banyak pengalaman yang berharga dalam memperdalam dan meningkatkan keilmuan dalam bidang informatika selama perkuliahan di Teknik Informatika ITS.

Tugas akhir ini selesai karena tidak lepas dari bantuan dan dukungan berbagai pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT. Atas segala rahmat dan berkatNya bagi penulis.
2. Bapak, Ibu, Mama, Papa dan Kakak penulis, Neni, Hakim, Humam, Nia, Halim yang memberikan doa, dukungan dan semangat untuk penulis
3. Khevin Putra Pratama yang setia menemani dan memberikan semangat dan dukungannya
4. Bapak Fajar Baskoro selaku dosen pembimbing 1 yang telah membantu dan membimbing penulis dalam menyelesaikan tugas akhir ini.
5. Ibu Adhatus Solichah selaku dosen pembimbing 2 yang telah membantu dan membimbing penulis dalam menyelesaikan tugas akhir ini.
6. Teman-teman grup *Gengs*, *Benalu*, *Fake*, *CGV* yang selalu menjadi penghibur dan teman nongkrong saat jenuh mengerjakan tugas akhir.
7. Teman-teman angkatan 2014 yang menjadi semangat bagi penulis selama masa perkuliahan di Teknik Informatika ITS.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juni 2018

Nurul Azizah

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxv
DAFTAR KODE SUMBER	xxvii
1 BAB I PENDAHULUAN	29
1.1. Latar Belakang	29
1.2. Tujuan.....	31
1.3. Rumusan Permasalahan.....	31
1.4. Batasan Permasalahan	31
1.5. Metodologi	31
1.6. Sistematika Penulisan.....	33
2 BAB II DASAR TEORI.....	35
2.1. Pariwisata	35
2.2. Generasi Millennial	35
2.3. Pariwisata di Era Millennial	36
2.4. Android.....	37
2.4.1. Activities	38
2.4.2. Services	38
2.4.3. Content Providers	38
2.4.4. Broadcast Receivers	38
2.4.5. SensorEventListener.....	39
2.4.6. LocationListener.....	39
2.4.7. Retrofit.....	39
2.5. Web Service REST API	40
2.6. JSON	41
2.7. MySQL.....	42
2.8. GPS (Global Positioning System)	44

2.9. PHP (Hypertext Preprocessor).....	46
2.10. Google Maps API.....	46
2.11. ReactJS	47
2.12. Perbandingan dengan Aplikasi Sejenis	49
3 BAB III ANALISIS DAN PERANCANGAN.....	51
3.1. Deskripsi Umum Sistem.....	51
3.1.1. Analisis Masalah	52
3.1.2. Analisis Kebutuhan Sistem	53
3.2. Perancangan Sistem.....	79
3.2.1. Perancangan Basis Data	79
3.2.2. Perancangan Web Service	85
3.2.3. Perancangan Antarmuka.....	87
3.2.4. Perancangan Proses Kontrol Komentar Negatif.....	91
4 BAB IV IMPLEMENTASI.....	93
4.1. Lingkungan Implementasi.....	93
4.1.1. Lingkungan Implementasi Perangkat Keras.....	93
4.1.2. Lingkungan Implementasi Perangkat Lunak.....	93
4.2. Lingkungan Implementasi Antarmuka	94
4.2.1. Antarmuka Splash Screen.....	94
4.2.2. Antarmuka Form Login pada Android	95
4.2.3. Antarmuka Form Register pada Android	95
4.2.4. Antarmuka Fitur Filter Kategori pada Android.....	96
4.2.5. Antarmuka List Hasil Filter Kategori pada Android.....	97
4.2.6. Antarmuka Detail Informasi Wisata pada Android ...	98
4.2.7. Antarmuka Fitur Review Destinasi Wisata pada Android.....	99
4.2.8. Antarmuka Melihat Profile Pengguna pada Android	100
4.2.9. Antarmuka Edit Profile Pengguna pada Android....	101

4.2.10. Antarmuka Halaman Add Gallery pada Android ..	102
4.2.11. Antarmuka Tambah Rekomendasi Wisata pada Android.....	103
4.2.12. Antarmuka Halaman Update Informasi Wisata pada Android.....	104
4.2.13. Antarmuka Kelola Wisata pada Web Administrator	105
4.2.14. Antarmuka Kelola Rekomendasi Wisata pada Web Administrator.....	105
4.2.15. Antarmuka Kelola Review Wisata pada Web Administrator.....	105
4.2.16. Antarmuka Kelola Profile Pengguna pada Web Administrator.....	106
4.3. Implementasi Basis Data	106
4.3.1. Implementasi Tabel Site	107
4.3.2. Implementasi Tabel Recommendation	107
4.3.3. Implementasi Tabel Review	108
4.3.4. Implementasi Tabel User.....	108
4.3.5. Implementasi Tabel Category.....	109
4.3.6. Implementasi Tabel Site_Category	109
4.3.7. Implementasi Tabel Gallery	109
4.4. Implementasi Web Service.....	110
4.5. Implementasi Proses Aplikasi	114
4.5.1. Implementasi Dependencies	114
4.5.2. Implementasi Pengiriman Data	116
4.5.3. Implementasi Pengambilan Data	117
5 BAB V PENGUJIAN DAN EVALUASI	119
5.1. Lingkungan Uji Coba	119
5.2. Uji Coba Fungsionalitas	119

5.2.1. Pengujian Fitur Registrasi	120
5.2.2. Pengujian Filter Kategori pada menu Home	121
5.2.3. Pengujian Melihat List hasil Filter Kategori	122
5.2.4. Pengujian Melihat destinasi wisata secara keseluruhan dalam bentuk list	124
5.2.5. Pengujian Melihat destinasi wisata secara keseluruhan dalam bentuk marker pada peta.....	124
5.2.6. Pengujian Melihat Detail Informasi wisata	126
5.2.7. Pengujian Memberikan review terkait destinasi yang dipilih	126
5.2.8. Pengujian Fitur Memberikan informasi terbaru terkait destinasi wisata yang sudah ada kepada administrator.....	128
5.2.9. Pengujian Fitur Memberikan rekomendasi wisata baru kepada administrator	129
5.2.10. Pengujian Melihat Profile.....	131
5.2.11. Pengujian Mengedit Profile.....	131
5.2.12. Pengujian Melakukan unggah foto untuk galeri spot wisata.....	133
5.2.13. Pengujian Mengelola konten wahana wisata.....	133
5.2.14. Pengujian Menanggapi rekomendasi pengguna	135
5.2.15. Pengujian Melakukan manajemen pengguna	136
5.2.16. Pengujian Kontrol komentar negatif	137
5.3. Uji Coba Response Time.....	139
5.3.1. Pengujian 100 Pengguna	140
5.3.2. Pengujian 500 Pengguna	140
5.3.3. Pengujian 1000 Pengguna	141
5.3.4. Pengujian 2000 Pengguna	141

5.3.5. Pengujian 3000 Pengguna	141
5.4. Evaluasi Pengujian	142
5.4.1. Evaluasi Pengujian Fungsionalitas	143
5.4.2. Evaluasi Pengujian Response Time.....	144
5.4.3. Evaluasi Pengujian Kualitas	144
6 BAB VI PENUTUP.....	149
6.1. Kesimpulan.....	149
6.2. Saran.....	149
DAFTAR PUSTAKA.....	151
BIODATA PENULIS.....	153

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Ide Dasar GPS Positioning	45
Gambar 3.1 Arsitektur Sistem	52
Gambar 3.2 Gambar Kasus Penggunaan	55
Gambar 3.3 Diagram Aktivitas Kasus Penggunaan UC-01	57
Gambar 3.4 Diagram Aktivitas Kasus Penggunaan UC-02	58
Gambar 3.5 Diagram Aktivitas Kasus Penggunaan UC-03	59
Gambar 3.6 Diagram Aktivitas Kasus Penggunaan UC-04	60
Gambar 3.7 Diagram Aktivitas Kasus Penggunaan UC-05	61
Gambar 3.8 Diagram Aktivitas Kasus Penggunaan UC-06	62
Gambar 3.9 Diagram Aktivitas Kasus Penggunaan UC-07	64
Gambar 3.10 Diagram Aktivitas Kasus Penggunaan UC-08	66
Gambar 3.11 Diagram Aktivitas Kasus Penggunaan UC-09	68
Gambar 3.12 Diagram Aktivitas Kasus Penggunaan UC-10	69
Gambar 3.13 Diagram Aktivitas Kasus Penggunaan UC-11	70
Gambar 3.14 Diagram Aktivitas Kasus Penggunaan UC-12	72
Gambar 3.15 Diagram Aktivitas Kasus Penggunaan UC-13	74
Gambar 3.16 Diagram Aktivitas Kasus Penggunaan UC-14	75
Gambar 3.17 Diagram Aktivitas Kasus Penggunaan UC-15	77
Gambar 3.18 Diagram Aktivitas Kasus Penggunaan UC-16	78
Gambar 3.19 CDM (Conceptual Data Model) pada web service	80
Gambar 3.20 PDM (Physical Data Model) pada web service	81
Gambar 3.21 Desain Antar Muka Form Login	87
Gambar 3.22 Desain Antar Muka Form Register	87
Gambar 3.23 Desain Antar Muka Fitur Filter Kategori	88
Gambar 3.24 Desain Antar Muka List hasil Filter Kategori	88
Gambar 3.25 Desain Antar Muka Marker Peta Destinasi Wisata	89
Gambar 3.26 Desain Antar Muka Detail Informasi Wisata	89
Gambar 3.27 Desain Antar Muka Fitur Review	90
Gambar 3.28 Desain Antar Muka Fitur Edit Profile	90
Gambar 3.29 Diagram Alir Proses Kontrol komentar negatif	91
Gambar 4.1 Tampilan Splash Screen	94
Gambar 4.2 Tampilan Form	95
Gambar 4.3 Tampilan Form	95

Gambar 4.4 Tampilan Fitur Filter Kategori pada Android.....	96
Gambar 4.5 Tampilan Card List Destinasi Wisata pada Android.....	97
Gambar 4.6 Tampilan Detail Informasi Wisata pada Android.....	98
Gambar 4.7 Tampilan Fitur Review Destinasi Wisata pada Android	99
Gambar 4.8 Tampilan Melihat Profile Pengguna pada Android.....	100
Gambar 4.9 Tampilan Edit Profile Pengguna pada Android.....	101
Gambar 4.10 Tampilan Halaman Add Gallery pada Android.....	102
Gambar 4.11 Tampilan Input Location pada fitur Rekomendasi Wisata pada Android.....	103
Gambar 4.12 Tampilan Tambah Rekomendasi Wisata pada Android.....	103
Gambar 4.13 Tampilan Halaman Update Informasi Wisata pada Android.....	104
Gambar 4.14 Tampilan Kelola Wisata pada Web Administrator	105
Gambar 4.15 Tampilan Kelola Review Wisata pada Web Administrator	106
Gambar 5.1 Registrasi pada Android	121
Gambar 5.2 Filter Kategori pada Android.....	122
Gambar 5.3 List hasil filter kategori pada Android.....	123
Gambar 5.4 List wisata secara keseluruhan pada Android.....	125
Gambar 5.5 Map wisata secara keseluruhan pada Android.....	125
Gambar 5.6 Detail Informasi wisata pada Android.....	127
Gambar 5.7 Review wisata pada Android.....	127
Gambar 5.8 Update Informasi Wisata pada Android.....	129
Gambar 5.9 Rekomendasi wisata baru pada Android	130
Gambar 5.10 Melihat profile	132
Gambar 5.11 Edit Profile pengguna pada Android	132
Gambar 5.12 Add Gallery pada Android	134
Gambar 5.13 Kelola Wisata pada Web Administrator	135
Gambar 5.14 Menanggapi rekomendasi wisata pada Web Administrator	136
Gambar 5.15 Manajemen pengguna pada Web Administrator	137

Gambar 5.16 Kontrol komentar negatif pada Web Administrator	138
Gambar 5.17 Rentan usia responden	143
Gambar 5.18 Hasil evaluasi uji kualitas tampilan sistem	145
Gambar 5.19 Hasil evaluasi uji kualitas penggunaan bahasa pada sistem.....	145
Gambar 5.20 Hasil evaluasi uji kualitas kemudahan penggunaan sistem.....	146
Gambar 5.21 Tingkat kepuasan responden	147

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Metode HTTP dan penggunaannya dalam REST Java Android.....	40
Tabel 2.2 Perbandingan Aplikasi sejenis.....	50
Tabel 3.1 Deskripsi Kasus Penggunaan	55
Tabel 3.2 Rincian Alur Kasus Penggunaan UC-01	56
Tabel 3.3 Rincian Alur Kasus Penggunaan UC-02	58
Tabel 3.4 Rincian Alur Kasus Penggunaan UC-03	59
Tabel 3.5 Rincian Alur Kasus Penggunaan UC-04	60
Tabel 3.6 Rincian Alur Kasus Penggunaan UC-05	61
Tabel 3.7 Rincian Alur Kasus Penggunaan UC-06	62
Tabel 3.8 Rincian Alur Kasus Penggunaan UC-07	63
Tabel 3.9 Rincian Alur Kasus Penggunaan UC-08	65
Tabel 3.10 Rincian Alur Kasus Penggunaan UC-09	67
Tabel 3.11 Rincian Alur Kasus Penggunaan UC-10	68
Tabel 3.12 Rincian Alur Kasus Penggunaan UC-11	69
Tabel 3.13 Rincian Alur Kasus Penggunaan UC-12	71
Tabel 3.14 Rincian Alur Kasus Penggunaan UC-13	73
Tabel 3.15 Rincian Alur Kasus Penggunaan UC-14	74
Tabel 3.16 Rincian Alur Kasus Penggunaan UC-15	76
Tabel 3.17 Rincian Alur Kasus Penggunaan UC-16	77
Tabel 3.18 Kebutuhan Non Fungsional.....	79
Tabel 3.19 Atribut Tabel Data User	81
Tabel 3.20 Atribut Tabel Data Recommendation.....	82
Tabel 3.21 Atribut Tabel Site	83
Tabel 3.22 Atribut Tabel Review	83
Tabel 3.23 Atribut Tabel Kategori	84
Tabel 3.24 Atribut Tabel Kategori_site.....	84
Tabel 3.25 Atribut Tabel Gallery	84
Tabel 3.26 Perancangan API Endpoint	85
Tabel 4.1 Lingkungan Implementasi Perangkat Keras.....	93
Tabel 4.2 Lingkungan Implementasi Perangkat Lunak.....	93
Tabel 4.3 Implementasi API Endpoint	110
Tabel 5.1 Pengujian Fitur Registrasi	120

Tabel 5.2 Pengujian Filtering pada fitur pencarian	121
Tabel 5.3 Pengujian Melihat list hasil filtering dalam bentuk card	122
Tabel 5.4 Pengujian Melihat destinasi wisata secara keseluruhan dalam bentuk list	124
Tabel 5.5 Pengujian Melihat destinasi wisata secara keseluruhan dalam bentuk peta.....	124
Tabel 5.6 Pengujian Melihat detail informasi wisata	126
Tabel 5.7 Pengujian Memberikan review terkait destinasi yang dipilih	126
Tabel 5.8 Pengujian Fitur Memberikan informasi terbaru terkait destinasi wisata yang sudah ada kepada administrator.....	128
Tabel 5.9 Pengujian fitur memberikan rekomendasi wisata baru kepada administrator	129
Tabel 5.10 Pengujian Melihat Profile.....	131
Tabel 5.11 Pengujian Mengedit Profile.....	131
Tabel 5.12 Pengujian Melakukan unggah foto untuk galeri spot wisata.....	133
Tabel 5.13 Pengujian Mengelola wahan wisata	133
Tabel 5.14 Pengujian Menanggapi rekomendasi pengguna	135
Tabel 5.15 Pengujian Melakukan manajemen pengguna	136
Tabel 5.16 Pengujian Kontrol komentar negatif	137
Tabel 5.17 Hasil Pengujian <i>Response Time</i> 100 Pengguna.....	140
Tabel 5.18 Hasil Pengujian <i>Response Time</i> 500 Pengguna.....	140
Tabel 5.19 Hasil Pengujian <i>Response Time</i> 1000 Pengguna.....	141
Tabel 5.20 Hasil Pengujian <i>Response Time</i> 2000 Pengguna.....	141
Tabel 5.21 Hasil Pengujian <i>Response Time</i> 3000 Pengguna.....	142
Tabel 5.22 Hasil Evaluasi uji fungsionalitas	143
Tabel 5.23 Perbandingan Hasil Pengujian <i>Response Time</i>	144
Tabel 5.24 Kritik dan saran responden.....	147

DAFTAR KODE SUMBER

Kode Sumber 2.1 Contoh penggunaan ReactJS	48
Kode Sumber 2.2 Hasil penggunaan ReactJS	48
Kode Sumber 4.1 Implementasi Tabel Site	107
Kode Sumber 4.2 Implementasi Tabel Recommendation	108
Kode Sumber 4.3 Implementasi Tabel Review	108
Kode Sumber 4.4 Implementasi Tabel Users	109
Kode Sumber 4.5 Implementasi Tabel Category	109
Kode Sumber 4.6 Implementasi Tabel Site_Category	109
Kode Sumber 4.7 Implementasi Tabel Gallery	110
Kode Sumber 4.8 Implementasi Dependencies Retrofit	115
Kode Sumber 4.9 Implementasi Dependencies Glide	115
Kode Sumber 4.10 Implementasi Dependencies Google Maps Service	115
Kode Sumber 4.11 Implementasi Dependencies ButterKnife...	115
Kode Sumber 4.12 Implementasi Pengiriman Data	116
Kode Sumber 4.13 Implementasi Mengambil Data	117

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1. Latar Belakang

Pariwisata merupakan aset sebuah daerah yang tidak ada habisnya. Salah satunya Magelang yang merupakan salah satu daerah yang memiliki potensi, begitu banyaknya destinasi wisata mulai dari wisata alam hingga wisata buatan serta aneka kuliner yang begitu beragam. Didukung oleh letak wilayah yang strategis, sumber daya alam yang melimpah, dan sumber daya manusia yang produktif, menjadikan Magelang berpotensi sebagai daerah tujuan wisata pilihan di Indonesia. Disamping itu, di era millennial atau kadang juga disebut dengan generasi Y, orang yang lahir pada kisaran tahun 1980- 2000an merupakan generasi yang memilih menghabiskan uang mereka untuk mendapatkan pengalaman (*experience*) dibandingkan barang (*material goods*) dan mayoritas pelaku kegiatan travelling adalah para kaum muda atau generasi millennial. Akan tetapi para wisatawan dirasa masih kesulitan dalam memilih destinasi wisata yang cocok dan sesuai keinginannya selain itu informasi *up to date* terkait destinasi wisata pun juga masih sulit didapat, dari permasalahan tersebut penulis tergerak untuk memunculkan sebuah inovasi yang dapat memenuhi dan tentunya memudahkan wisatawan, dilihat dari kecenderungan pengguna yang rata-rata adalah era millennial, maka teknologi perangkat bergerak dirasa merupakan solusi terbaik.

Teknologi perangkat bergerak (*mobile device*) menjadi salah satu solusi untuk memunculkan inovasi baru, salah satu nya dalam memenuhi kebutuhan akan wisata dan dalam usaha mengefektifkan serta meningkatkan usaha pemasaran industri pariwisata. Oleh karenanya Aplikasi Panduan Wisata menjadi

solusinya, meskipun aplikasi panduan wisata di daerah Magelang sudah ada sebelumnya akan tetapi fitur yang ada dirasa masih kurang optimal, dari sini penulis mengusulkan sebuah aplikasi panduan wisata, yang mana aplikasi ini nantinya dapat mengelompokkan tempat wisata yang ada di Magelang berdasarkan kebutuhan pengguna era millennial, dimana wisatawan disajikan list kriteria seperti apa wisata yang diinginkan, dengan menggunakan fitur *filter* kriteria yang kemudian akan ditampilkan list destinasi wisata yang sesuai dengan kriteria wisatawan. Tujuan kedua yaitu menyajikan informasi lengkap dan *up to date* terkait wisata yang dipilih, selain itu pengguna juga dapat melihat dan memberikan ulasan terkait destinasi wisata yang dipilih, untuk mekanisme pengumpulan data wisata dilakukan melalui input manual administrator dimana pengumpulan data ini berdasarkan kontribusi masyarakat yang mana pengguna dapat mengirim rekomendasi wisata melalui *platform* Android yang nantinya akan dicek ulang pada halaman *website* administrator sebelum diunggah, tak hanya itu, pengguna juga dapat memberikan *update* informasi terbaru terkait wisata yang sudah ada sebelumnya kepada administrator serta pengguna juga dapat mengunggah foto spot wisata pada galeri yang disediakan pada aplikasi.

Diharapkan dengan diterapkannya solusi ini, Wisatawan dapat dengan mudah menemukan destinasi wisata yang sesuai dengan kriteria yang diinginkannya berdasar kebutuhan pengguna era millennial, begitupula Dinas Pariwisata Magelang yang akan terbantu dalam promosi wisata Magelang dan juga dalam memperbaiki maupun meningkatkan kualitas wisata dari hasil *review* pengunjung pada Aplikasi ini nantinya.

1.2. Tujuan

Pembuatan tugas akhir ini bertujuan untuk menyajikan sebuah aplikasi yang sesuai dengan kebutuhan pengguna era millennial serta mempermudah wisatawan (pengguna) untuk mengetahui informasi lengkap dan *up to date* terkait wisata di Magelang sesuai kriteria yang diinginkannya.

1.3. Rumusan Permasalahan

Rumusan masalah pada pembuatan tugas akhir ini adalah:

1. Bagaimana mekanisme pengambilan data wisata dari basis data menggunakan konsep REST?
2. Bagaimana mekanisme pengawasan administrator terhadap komentar negatif?
3. Bagaimana perancangan dan implementasi aplikasi panduan wisata yang sesuai dengan pengguna era millennial pada perangkat bergerak Android?

1.4. Batasan Permasalahan

Asumsi dan ruang lingkup permasalahan yang akan dibahas pada tugas akhir ini adalah:

1. Teknologi yang digunakan dalam pembuatan aplikasi ini adalah bahasa pemrograman Android Java, kerangka kerja ReactJS, serta web service REST API.
2. Aplikasi untuk perangkat bergerak dikhususkan untuk perangkat dengan sistem operasi android.
3. Aplikasi untuk manajemen konten dibuat dengan basis web.

1.5. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini yaitu:

1. Studi literatur

Tahap ini merupakan proses pengumpulan informasi yang dibutuhkan untuk mengerjakan tugas akhir. Informasi yang dibutuhkan dapat berupa literatur dan dokumentasi

penggunaan. Studi literatur yang akan dipelajari untuk tugas akhir ini adalah sebagai berikut:

1. Mekanisme penggunaan GPS untuk menampilkan lokasi wisata pada jarak yang terdekat.
2. Mekanisme filtering pada pencarian destinasi wisata.
3. Menampilkan destinasi wisata berupa pin pada peta digital.
4. Mekanisme kontrol komentar negatif

2. Analisis dan Desain Perangkat Lunak

Tahap ini dilakukan analisis terhadap sistem serta perancangan sistem yang akan dibuat. Tujuannya adalah untuk merumuskan solusi dalam pelaksanaan implementasi pada sistem. Secara garis besar, fitur utama yang terdapat pada program ini adalah:

1. Filter kriteria destinasi wisata
2. Menampilkan destinasi wisata

3. Implementasi

Untuk implementasi aplikasi panduan wisata, implementasi pada tugas akhir ini adalah sebagai berikut:

1. Implementasi Aplikasi Web
Aplikasi untuk administrator mengelola *database* wisata melalui REST API yang tersedia pada aplikasi web sebagai konten pada Aplikasi android.
2. Implementasi *Server*
Pada *server* akan digunakan basis data MySQL dan bahasa PHP untuk membuat REST API.
3. Implementasi Aplikasi Android
Aplikasi untuk panduan wisata Magelang diimplementasikan pada *smartphone* Android menggunakan bahasa pemrograman Java.

4. Pengujian dan evaluasi

Pengujian untuk tugas akhir ini sebagai berikut:

1. Melakukan uji coba fungsional apakah semua fitur telah berjalan dengan baik.
2. Melakukan ujicoba kualitas dan tingkat kepuasan pengguna apakah aplikasi sudah sesuai dengan kebutuhan pengguna.
3. Pada peta digital akan menampilkan destinasi wisata berupa pin pada peta digital.

5. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan proses dokumentasi dan pembuatan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir.

1.6. Sistematika Penulisan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan tugas akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan tugas akhir ini.

Bab III Metode Pemecahan Masalah

Bab ini membahas mengenai Metode yang digunakan untuk memecahkan masalah yang dipaparkan pada rumusan permasalahan.

Bab IV Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur dan proses.

Bab V Implementasi

Bab ini berisi implementasi dari perancangan dan implementasi dalam bentuk coding. Bab ini berisi proses pembangunan perangkat lunak

Bab VI Pengujian dan Evaluasi

Bab ini membahas tentang pengujian aplikasi berdasarkan skenario yang telah ditentukan. Mengevaluasi hasil uji coba dari perangkat lunak.

Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dari proses pengembangan perangkat lunak dan hasil uji coba.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah atau kode-kode sumber yang penting pada aplikasi

BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan tugas akhir.

2.1. Pariwisata

Pariwisata atau turisme adalah suatu perjalanan yang dilakukan untuk rekreasi atau liburan dan juga persiapan yang dilakukan untuk aktivitas ini. Seorang wisatawan atau turis adalah seseorang yang melakukan perjalanan paling tidak sejauh 80 km (50 mil) dari rumahnya dengan tujuan rekreasi, merupakan definisi oleh Organisasi Pariwisata Dunia.

Definisi yang lebih lengkap, turisme adalah industri jasa. Mereka menangani jasa mulai dari transportasi, jasa keramahan, tempat tinggal, makanan, minuman dan jasa bersangkutan lainnya seperti bank, asuransi, keamanan dll. Dan juga menawarkan tempat istirahat, budaya, pelarian, petualangan, pengalaman baru dan berbeda lainnya.

Banyak negara bergantung banyak dari industri pariwisata ini sebagai sumber pajak dan pendapatan untuk perusahaan yang menjual jasa kepada wisatawan. Oleh karena itu pengembangan industri pariwisata ini adalah salah satu strategi yang dipakai oleh Organisasi Non-Pemerintah untuk mempromosikan wilayah tertentu sebagai daerah wisata untuk meningkatkan perdagangan melalui penjualan barang dan jasa kepada orang non-lokal. Menurut Undang Undang No. 10/2009 tentang Kepariwisataaan, yang dimaksud dengan pariwisata adalah berbagai macam kegiatan wisata yang didukung oleh berbagai fasilitas serta layanan yang disediakan masyarakat, pengusaha, Pemerintah dan Pemerintah Daerah [1].

2.2. Generasi Millennial

Generasi millennial atau kadang juga disebut dengan generasi Y adalah sekelompok orang yang lahir setelah Generasi X, yaitu orang yang lahir pada kisaran tahun 1980- 2000an. Maka berarti

millennials adalah generasi muda yang berusia 18-38 tahun di tahun ini. Dibanding generasi sebelum, generasi millennial memang unik, hasil riset yang dirilis oleh Pew Research Center misalnya secara gamblang menjelaskan keunikan generasi millennial dibanding generasi-generasi sebelumnya adalah soal penggunaan teknologi dan budaya. Kehidupan generasi millennial tidak bisa dilepaskan dari teknologi terutama internet, *entertainment*/hiburan sudah menjadi kebutuhan pokok bagi generasi ini. [2]

2.3. Pariwisata di Era Millennial

Berdasarkan hasil survei di seluruh dunia yang dirilis oleh *Everbrite-Harris Poll* 2014, generasi millennial lebih memilih menghabiskan uang mereka untuk mendapatkan pengalaman (*experience*) dibandingkan barang (*material goods*). Peluang inilah yang tidak disia-siakan oleh Kementerian Pariwisata untuk membangun destinasi digital yang berkonsep kekinian guna memaksimalkan perjalanan wisata di Indonesia. [3]

Selanjutnya seperti diungkapkan *Senior Managing Director Products, Digital Customer Lead, Accenture Consulting*, Simon Eaves, milenial menginginkan sebuah kemudahan dan juga pengguna teknologi digital. "Mereka menginginkan kemudahan dengan teknologi digital dari ujung ke ujung. Ingin ada hubungan mulai dari baru keluar rumah, melakukan perjalanan, dan bahkan sampai ke rumah lagi. Jadi dari mulai proses check-in di bandara, lalu bisa ke mana saja, hotel, dan seterusnya," kata Simon. Oleh karena itu, lanjut Simon, brand industri pariwisata perlu memanfaatkan teknologi digital.

Simon menambahkan salah satu karakter milenial adalah ingin sesuatu yang unik dan terpersonalisasi sesuai karakter masing-masing.

Sementara itu *Managing Director Products Accenture Consulting*, Yap Boom Lim mengatakan bahwa berdasarkan survei yang mereka lakukan, lebih dari 50 persen milenial menggunakan peralatan digital untuk berinteraksi dengan hotel. "Mereka menggunakannya dari sejak perjalanan hingga seluruh perjalanan

dan akhir perjalanan. Ini konsisten hasil surveinya, tidak pandang negara mana, hal ini terjadi di seluruh dunia," kata Lim. Hal ini menunjukkan milenial menggunakan teknologi digital untuk mengatur perjalanan mereka. Lim mengatakan karakter milenial senang melakukan segala sesuatunya sendiri, mengutamakan efisiensi dan kemudahan, serta kurang sabar. Oleh karena itu, lanjut Lim, tak heran teknologi digital memegang peranan penting terhadap perilaku milenial. [4]

2.4. Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler [5]. Pada tugas akhir ini, aplikasi pendeteksi *bump* akan dibangun pada *smartphone* dengan sistem operasi Android. Aplikasi ini akan dibangun menggunakan Android Studio.

Aplikasi Android menggunakan bahasa pemrograman Java. Android SDK (Software Development Kit) sebagai *compiler* kode aplikasi. Komponen aplikasi adalah bagian penting dalam pengembangan aplikasi Android. Komponen dapat berupa *entry point*, yaitu kontrol penghubung antara sistem operasi dengan program aplikasi. Setiap komponen memiliki ketergantungan dengan komponen yang lain. Berikut ini adalah 4 tipe komponen aplikasi [6].

2.4.1. Activities

Activity merepresentasikan sebuah tampilan pada antarmuka pengguna. Misalnya, aplikasi email memiliki satu *activity* yaitu untuk membuat email. Namun pada satu aplikasi dapat terdapat beberapa *activity* yang bekerjasama untuk membuat aplikasi yang dapat digunakan oleh *user* dengan mudah. Sebuah *activity* dapat diimplementasikan menggunakan subclass *Activity*.

2.4.2. Services

Komponen *service* merupakan komponen yang bekerja pada latarbelakang untuk menjalankan sebuah operasi. *Service* tidak menyediakan antarmuka pengguna. Contoh penggunaan ini adalah menjalankan musik dilatarbelakang (*background*). Sebuah *service* dapat diimplementasikan menggunakan subclass *Services*.

2.4.3. Content Providers

Content provider berguna untuk mengatur seperangkat data aplikasi. Data dapat disimpan dalam sistem, SQLite, dan penyimpanan persisten lainnya. Melalui *content provider*, aplikasi lain dapat mengambil atau memodifikasi data. *Content provider* berguna untuk membaca dan menulis data yang rahasia dan tidak boleh dibagikan, misalnya aplikasi notepad menggunakan *content provider* untuk menyimpan catatan.

2.4.4. Broadcast Receivers

Broadcast receiver adalah komponen yang merespon pengumuman *broadcast* ke semua sistem. Contoh *broadcast* yang berasal dari sistem adalah *broadcast* yang muncul ketika layar telah dimatikan, baterai sedikit, dan lain lain. Aplikasi ini juga dapat memulai *broadcast*, misalnya memberitahu aplikasi lain bahwa data telah diunduh dan siap untuk digunakan. Pada umumnya *broadcast receiver* berguna sebagai pintu gerbang untuk komponen lain dan dimaksudkan untuk melakukan jumlah pekerjaan yang sedikit.

2.4.5. SensorEventListener

SensorEventListener merupakan antarmuka publik yang terdapat pada Android yang berfungsi untuk menerima pemberitahuan pembaharuan data dari *sensorManager*. *Public Method* yang digunakan ada 2 yaitu *onAccuracyChanged* dan *onSensorChanged*. *Public Method onAccuracyChanged* adalah *listener* aplikasi untuk perubahan akurasi pada sensor yang sedang digunakan, sedangkan *onSensorChanged* adalah *listener* perubahan nilai sensor.

2.4.6. LocationListener

LocationListener [7] adalah penghubung antara aplikasi dengan GPS. *LocationListener* terdiri dari *LocationManager* dan *LocationListener*. *LocationManager* adalah layanan akses lokasi pada Android. Salah satu layanan yang tersedia adalah memperbarui posisi lokasi dimana *smartphone* tersebut sedang digunakan. *Method* ini dipanggil jika *LocationListener* telah terdaftar dengan *LocationManager* menggunakan *requestLocationUpdates()*. Berikut adalah *Public Method* yang digunakan.

1. *onLocationChanged*, fungsi ini dipanggil ketika lokasi berubah. Nilai garis bujur dan garis lintang akan diperbarui ketika fungsi ini dipanggil dalam bentuk objek *LocationListener*.
2. *onProviderDisabled*, digunakan ketika provider dinon-aktifkan oleh pengguna.
3. *onProviderEnabled*, digunakan ketika provider diaktifkan oleh pengguna.
4. *onStatusChanged*, dipanggil saat ada perubahan status pada provider.

2.4.7. Retrofit

Retrofit adalah *library REST Client (Helper Library)* yang digunakan di Android dan Java untuk membuat *request HTTP* dan juga untuk memproses respon *HTTP* dari *REST API*. Ini dibuat oleh

Square, retrofit dapat juga digunakan untuk menerima struktur data selain *JSON*, misalnya *SimpleXML* dan *Jackson*.

REST Client adalah *library Retrofit* yang digunakan di sisi klien (Android) untuk membuat *request HTTP* ke *REST API*.

A *REST API* mendefinisikan sekumpulan fungsi yang dapat dilakukan pengembang untuk melakukan permintaan dan menerima tanggapan melalui protokol *HTTP* seperti *GET* dan *POST*. *RESTful API* adalah antarmuka program aplikasi (*API*) yang menggunakan permintaan *HTTP* untuk *GET*, *PUT*, *POST*, dan *DELETE* data. [8]

2.5. Web Service REST API

REST (Representational State Transfer) merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Umumnya menggunakan HTTP (Hypertext Transfer Protocol) sebagai protocol untuk komunikasi data. Pada arsitektur REST, REST server menyediakan *resources* (sumber daya/data) dan REST *client* mengakses dan menampilkan *resource* tersebut untuk penggunaan selanjutnya. Setiap *resource* diidentifikasi oleh URI (Universal Resource Identifiers) atau global ID. Resource tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML [9].

Tabel 2.1 Metode HTTP dan penggunaannya dalam REST Java Android

Metode	Deskripsi
GET	Mendapatkan (<i>read</i>) sebuah sumber daya (<i>resource</i>) yang diidentifikasi dengan URI (<i>Uniform Resource Identifier</i>)
POST	Mengirimkan sumber daya (<i>resource</i>) ke server. Digunakan untuk membuat (<i>create</i>) sumber daya baru

Metode	Deskripsi
PUT	Mengirimkan sumber daya (<i>resource</i>) ke server. Digunakan untuk memasukkan (<i>insert</i>) atau memperbarui (<i>update</i>) sumber daya yang tersimpan.
DELETE	Menghapus (<i>delete</i>) sumber daya (<i>resource</i>) yang diidentifikasi dengan URI
HEAD	Mendapatkan metadata (<i>response header</i>) dari sumber daya (<i>resource</i>) yang diidentifikasi dengan URI.

Arsitektur REST yang *decoupled* (terpisah) serta beban komunikasi yang ringan antara produsen dan konsumen membuatnya populer di dunia *cloud-based API*, seperti yang disajikan oleh Amazon, Microsoft, dan Google. Layanan berbasis *web* yang menggunakan arsitektur REST semacam itu dinamakan RESTful APIs (*Application Programming Interfaces*) atau REST API [10].

2.6. JSON

JSON adalah singkatan dari Java Script Object Notation, yaitu sebuah format untuk pertukaran data. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. JSON terbuat dari dua struktur:

- Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
- Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik

(*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini [11].

2.7. MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional,

hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional [12] [13]. Keistimewaan MySQL:

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka (open source). MySQL didistribusikan sebagai open source sehingga dapat digunakan secara gratis.
3. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. Performance tuning, MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti password yang terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah record lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau named pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh

bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Antar Muka. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle..

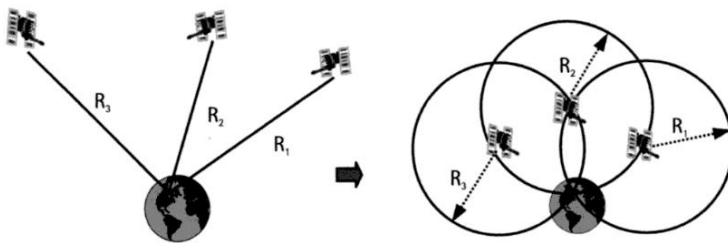
2.8. GPS (Global Positioning System)

GPS adalah sistem navigasi berbasis ruang yang menyediakan informasi lokasi dan waktu dalam segala kondisi cuaca, dimanapun di bumi yang tak terhalang pandangan 4 atau lebih satelit GPS. Sistem ini menggunakan 24 satelit yang mengirimkan gelombang mikro ke bumi. Sinyal ini kemudian diterima oleh alat penerima di permukaan bumi. Data yang diterima itu kemudian digunakan untuk menentukan posisi, kecepatan, arah dan waktu. Satelit GPS mengitari bumi secara sirkular (mendekati bentuk elips) dengan inklinasi sebesar 55 derajat [14].

GPS pada awalnya dikelola oleh *US Air Force* (USAF) sebagai operator sistem. GPS yang pada awalnya dikembangkan untuk sistem kekuatan tambahan bagi militer memiliki potensi menguntungkan bagi masyarakat sipil. Pada akhirnya GPS menyediakan dua layanan, yaitu layanan *Precise Positioning Service* (PPS) yang berguna untuk kekuatan militer AS dan layanan *Standard Positioning Service* (SPS) yang didesain untuk memberikan kemampuan *positioning*.

GPS memiliki 3 segmen, yaitu segmen angkasa, segmen kontrol, dan segmen pengguna. Segmen angkasa terdiri dari 24-

satelit yang telah dijelaskan sebelumnya. Setiap satelit GPS secara terus menerus akan mentransmisikan gelombang sinyal radio mikro yang terdiri dari dua *carriers*, dua kode dan pesan navigasi. Ketika GPS *receiver* diaktifkan, GPS *receiver* akan mengambil sinyal melalui antena penerima. Setelah menerima sinyal GPS, sinyal tersebut akan diproses menggunakan software yang ada didalam perangkat tersebut. Secara teori hanya dibutuhkan informasi tiga jarak antara GPS *receiver* dengan satelit, artinya dibutuhkan sebanyak minimal tiga satelit yang melacak secara bersamaan. Pada kasus ini, penerima berada pada persimpangan dari tiga lingkup; masing-masing memiliki jarak radius seperti ditunjukkan pada Gambar 2.1



Gambar 2.1 Ide Dasar GPS Positioning

Pada segmen kontrol sistem GPS terdiri dari stasiun pelacak yang tersebar pada jaringan seluruh dunia, dengan kontrol utamanya berada di negara Amerika Serikat di Colorado Springs, Colorado. Tujuan utama segmen kontrol adalah melacak satelit GPS dengan maksud untuk menentukan dan memprediksi lokasi satelit, integritas satelit, perilaku satelit, jam atom, data atmosfer dan lain-lain. Pada segmen pengguna sendiri terdiri dari seluruh bagian militer dan pengguna sipil. Dengan adanya GPS *receiver* yang tersambung pada antena GPS, pengguna dapat menerima sinyal GPS, yang dapat digunakan untuk menentukan posisi dirinya dimanapun ia berada. GPS sekarang ini tersedia kepada semua pengguna diseluruh dunia dengan gratis.

2.9. PHP (Hypertext Preprocessor)

PHP merupakan bahasa pemrograman *server-side* yang digunakan untuk pengembangan *web* dan bahasa yang digunakan secara luas dalam pengembangan *website*. PHP dikembangkan pertama kali oleh Rasmus Lerdorf pada tahun 1995. Secara resmi dirilis pada tahun 1997. Fungsionalitas dasar pemrograman berorientasi objek ditambahkan pada PHP 3 dan ditingkatkan pada PHP 4. PHP telah digunakan lebih dari 200 juta *website* yang berbeda, dari *website* personal biasa sampai perusahaan raksasa seperti Facebook, Wikipedia, Tumblr, Slack dan Yahoo.

PHP [15] didesain untuk membuat halaman *web* yang dinamis. PHP berguna sebagai bahasa pemrograman scripting yang dikhususkan untuk pengembangan *web server-side*. PHP memiliki peran utama sebagai filter, yaitu mengambil *file* atau *stream* yang mengandung teks dan atau instruksi PHP dan menghasilkan data *stream* lain. PHP dijalankan pada komputer untuk digunakan oleh seorang pengguna, biasanya dijalankan pada *web server* dan diakses oleh banyak pengguna yang menggunakan *web browser*. PHP memiliki kelebihan yaitu menyembunyikan kompleksitas. Pada tugas akhir ini, PHP akan digunakan untuk membangun aplikasi *web service*.

2.10. Google Maps API

Google Maps API [16] adalah layanan oleh Google yang dapat digunakan untuk menampilkan data dalam bentuk peta digital melalui beberapa model fitur. Google Maps API menyediakan *web service* sebagai antarmuka untuk permintaan data Maps API dari layanan luar dan digunakan pada aplikasi Maps pengguna. Layanan *web* ini menggunakan HTTP *request* ke URL yang spesifik, meneruskan parameter URL sebagai argumen ke layanan.

Pada umumnya, layanan ini mengembalikan data dengan HTTP *request* berupa JSON atau XML untuk diuraikan dan atau

diproses oleh aplikasi. Fitur *web service* Google Maps API adalah sebagai berikut.

1. *Direction* API
2. *Distance Matrix* API
3. *Elevation* API
4. *Geolocation* API
5. *Road* API
6. *Time Zone* API
7. *Places* API

2.11. ReactJS

Kerangka kerja dengan bentuk arsitektur Model-View-Controller membagi aplikasi dalam tiga komponen utama, yaitu Model, View dan Controller. ReactJS merupakan library Javascript untuk membuat view/antarmuka pengguna. ReactJS dikembangkan oleh Facebook, Instagram serta komunitas dari berbagai pengembang individu atau korporasi. React dapat digunakan untuk melakukan handling layer View pada konsep MVC untuk aplikasi web maupun aplikasi mobile. ReactJS menonjolkan keunggulan dalam hal kecepatan, kesederhanaan, dan skalabilitasnya.

Dalam ReactJS, kode HTML dipecah menjadi komponen kecil-kecil. Kode yang telah dipecah-pecah tersebut dapat digunakan kembali pada berbagai tempat lainnya, dengan state dan properties yang berbeda. Setiap komponen dapat mengandung komponen lainnya. Semua komponen dalam ReactJS state nya dapat berubah sewaktu waktu, dan ReactJS lah yang akan melakukan update kedalam komponen ketika state nya berganti.

Fitur-fitur dalam ReactJS

1. JSX - JavaScript syntax extension
2. Components
3. Unidirectional data flow and Flux

Keunggulan ReactJS

1. Menggunakan virtual DOM yang merupakan sebuah objek JavaScript. Hal inilah yang menjadi keunggulan dari ReactJS

terutama dalam hal performa. JavaScript virtual DOM lebih cepat daripada regular DOM.

2. Dapat digunakan pada sisi client maupun sisi server bersamaan dengan frameworks lainnya.

3. Component dan data patterns mudah untuk dibaca, dimana membantu saat melakukan pengembangan aplikasi.

Kelemahan ReactJS

1. Hanya mencakup layer View pada arsitektur MVC(Model-View-Controller), sedangkan layer Model dan Controller harus dicakup menggunakan kerangka lainnya.

2. Dapat menggunakan JSX maupun kode HTML pada baris program, sehingga dapat terasa aneh bagi beberapa developer.

Berikut adalah contoh sederhana penggunaan ReactJS.

```
<div id="myReactApp"></div>

<script type="text/babel">
  class Greeter extends React.Component {
    render() {
      return <h1>{this.props.greeting}</h1>
    }
  }

  ReactDOM.render(<Greeter    greeting="Hello    World!"    />,
document.getElementById('myReactApp'));
</script>
```

Kode Sumber 2.1 Contoh penggunaan ReactJS

Dengan hasil sebagai berikut:

```
<div id="myReactApp">
  <h1>Hello World!</h1>
</div>
```

Kode Sumber 2.2 Hasil penggunaan ReactJS

2.12. Perbandingan dengan Aplikasi Sejenis

Dilakukan perbandingan antara beberapa aplikasi sejenis yang sudah ada sebelumnya pada *playstore*. Dapat dilihat pada Tabel 2.2 bahwa fitur pada aplikasi Magelang Tour lebih unggul bila dibandingkan 6 aplikasi panduan wisata yang sudah ada sebelumnya yaitu aplikasi Wisata Kota Magelang, Wisata Magelang, Magelang Cerdas, Borobudur Temple, Pariwisata Magelang dan Pesona Magelang, fitur yang dibandingkan meliputi: filter kriteria, pencarian, registrasi pengguna, kelola profile, review wisata, informasi lokasi wisata berupa *pin map*, jarak lokasi wisata dengan pengguna, *upload* foto untuk galeri spot wisata yang dapat dipilih untuk *upload public* atau *private*, memberi rekomendasi wisata kepada administrator baik itu wisata baru maupun *update* informasi wisata yang sudah ada sebelumnya dan yang terakhir deskripsi wisata.

Tabel 2.2 Perbandingan Aplikasi sejenis

No	Fitur		Magelang Tour	Wisata Kota Magelang	Wisata Magelang	Magelang Cerdas	Borobudur Temple	Pariwisata Magelang	Pesona Magelang
1	Filter Kriteria		✓						
2	Pencarian		✓	✓	✓	✓			✓
3	Registrasi		✓			✓			✓
4	Kelola Profil		✓						
5	Review	Rating	✓			✓			✓
6		Komentar	✓			✓			✓
7	Lokasi wisata pada peta		✓		✓	✓			✓
8	Memberi Rekomendasi	Wisata baru	✓						
9		Update Info	✓						
10	Deskripsi Wisata		✓	✓	✓	✓		✓	✓
11	Jarak Lokasi Otomatis		✓			✓			✓
12	Galeri Foto	Pribadi	✓						
13		Umum	✓						

BAB III

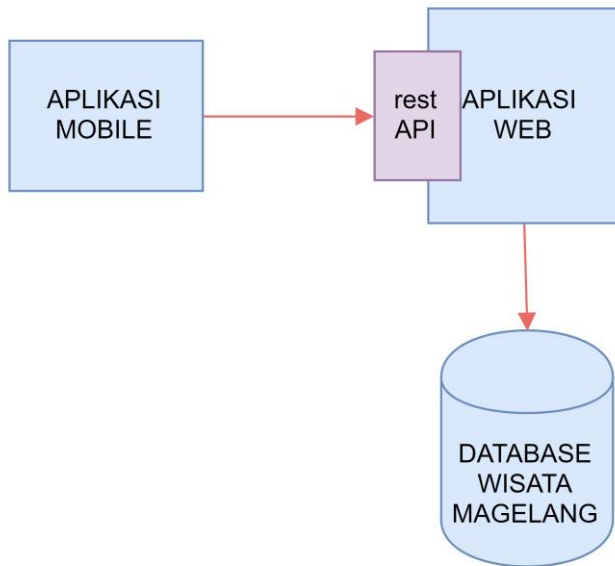
ANALISIS DAN PERANCANGAN

Pada bab ini akan membahas tahap analisis dan perancangan sistem yang akan dibangun pada tugas akhir. Perancangan meliputi perancangan data, perancangan proses, perancangan sistem dan perancangan antarmuka perangkat lunak

3.1. Deskripsi Umum Sistem

Sistem yang dibangun pada tugas akhir ini merupakan suatu sistem yang terdiri dari aplikasi perangkat bergerak berbasis Android dan aplikasi berbasis web. Aplikasi *mobile* berfungsi untuk mengambil konten dari *database* yang berisi daftar destinasi wisata melalui REST API yang tersedia pada aplikasi web.

Aplikasi ini nantinya dapat mengelompokkan tempat wisata yang ada di Magelang berdasarkan tren era anak millennial, dimana wisatawan disajikan list kriteria seperti apa wisata yang diinginkan, dengan menggunakan fitur *filter* kategori yang kemudian nantinya akan ditampilkan list destinasi wisata yang sesuai dengan kriteria wisatawan. Tujuan kedua yaitu menyajikan informasi lengkap dan *up to date* terkait wisata yang dipilih, selain itu pengguna juga dapat melihat dan memberikan review terkait destinasi wisata yang dipilih, tak hanya itu, pengguna juga dapat memberikan rekomendasi wisata baru ataupun *update* info terbaru terkait wisata yang sudah ada kepada administrator yang nantinya akan di cek ulang oleh administrator sebelum diunggah, sedangkan pada sisi administrator, administrator berfungsi untuk mengelola konten wisata baik itu menambah, mengubah ataupun menghapus, selain itu administrator juga berfungsi dalam manajemen pengguna, menanggapi rekomendasi wisata dari pengguna dan yang terakhir administrator berfungsi sebagai kontroler terhadap komentar negatif. Alur sistem secara keseluruhan atau arsitektur sistem ditampilkan pada Gambar 3.1



Gambar 3.1 Arsitektur Sistem

3.1.1. Analisis Masalah

Dalam proses pengambilan data wisata dari database, digunakan konsep REST API, dimana data dari aplikasi web direpresentasikan dalam bentuk format JSON yang nantinya akan dibaca pada aplikasi mobile. Selanjutnya mekanisme pengumpulan data wisata dilakukan melalui *input* manual administrator dan rekomendasi pengguna. Pengawasan atas komentar negatif juga dilakukan oleh administrator dengan cara pembuatan bank data dimana bank data tersebut berisi kata-kata yang tidak boleh ditampilkan pada kolom *review*, sehingga ketika ada pengguna yang *mensubmit* komentar dan mengandung kata-kata yang terdapat pada bank data maka sistem akan memberikan pemberitahuan kepada pengguna bawasannya ulasan yang diberikan mengandung kata-kata yang tidak seharusnya yang secara otomatis sistem tidak akan *mensubmit* ulasan tersebut.

3.1.2. Analisis Kebutuhan Sistem

Pada subbab ini akan dibahas secara mendalam tentang kebutuhan fungsional sistem dan kebutuhan non fungsional sistem yang diperlukan pada tugas akhir ini.

3.1.2.1. Penggalan Kebutuhan

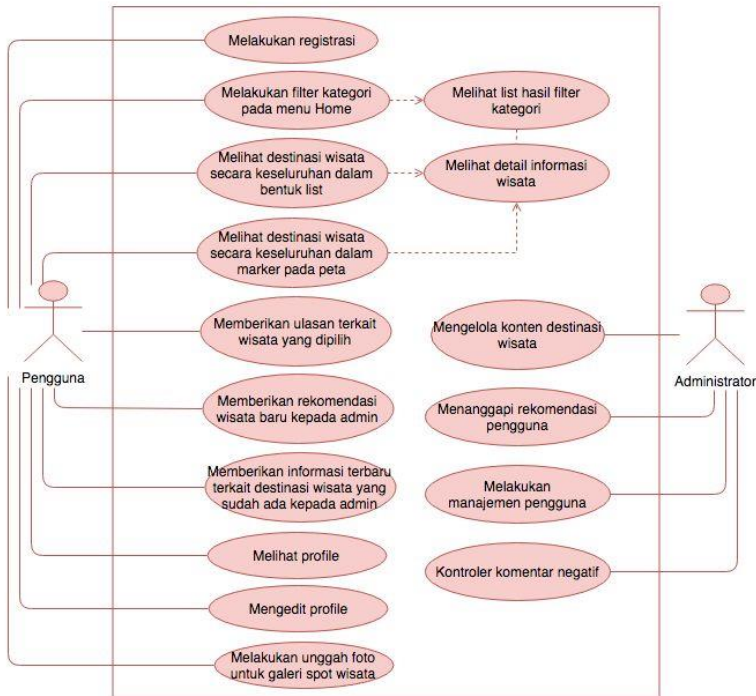
Pemilihan fitur – fitur pada aplikasi Magelang our didasarkan pada hasil perbandingan dengan aplikasi sejenis sesuai pada Tabel 2.2 pada bab 2 serta dari hasil survei pakar, dimana aplikasi Magelang tour mengambil fitur-fitur yang terdapat pada aplikasi panduan wisata yang sudah ada sebelumnya, selain itu aplikasi magelang tour ini juga menambahkan fitur baru seperti fitur kriteria, memberi rekomendasi wisata, memberikan *update* informasi wisata dan galeri foto. Penambahan fitur baru ini didasarkan dari survei yang dilakukan *Managing Director Products, Digital Customer Lead, Accenture Consulting*, Simon Eaves bawasannya salah satu karakter milenial adalah ingin sesuatu yang unik dan terpersonalisasi sesuai karakter masing-masing, dari sini lah fitur filter kriteria dirasa menjadi solusi untuk mempersonalisasikan destinasi wisata seperti apa yang diinginkan oleh pengguna dan fitur unggah foto pada galeri menjadi daya tarik keunikan aplikasi yang menjadikan ini solusi bagi pengguna untuk mengekspresikan diri dan membagikannya ke pengguna lain dimana dapat dilihat bawasannya kecenderungan pengguna jaman sekarang yang suka membagikan (*share*) kegiatan dalam bentuk foto. Selain itu fitur memberi rekomendasi serta memberikan *update* informasi wisata ditujukan untuk mempermudah administrator dalam mendapatkan informasi ataupun menambahkan data wisata baru. Selanjutnya pengkategorian filter wisata didapat dari penggalan informasi dari beberapa sumber site dimana kategori wisata dikelompokkan menjadi beberapa jenis, yang secara umum terbagi menjadi wisata alam, wisata religi, wisata belanja, wisata kuliner, wisata edukasi, wisata budaya, wisata agrowisata dan wisata petualangan.

3.1.2.2. Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang memiliki keterkaitan langsung dengan sistem. Kebutuhan fungsional pada sistem sebagai berikut:

- 1) Kebutuhan pengguna
 - Melakukan registrasi
 - Melakukan filter kategori pada menu Home
 - Melihat list hasil filter kategori
 - Melihat destinasi wisata secara keseluruhan dalam bentuk list
 - Melihat destinasi wisata secara keseluruhan dalam bentuk marker pada peta
 - Melihat detail informasi wisata
 - Memberikan ulasan terkait wisata yang dipilih
 - Memberikan informasi terbaru terkait destinasi wisata yang sudah ada kepada admin
 - Memberikan rekomendasi wisata baru kepada administrator
 - Melihat profile
 - Mengedit profile
 - Melakukan unggah foto untuk galeri spot wisata
- 2) Kebutuhan administrator
 - Mengelola konten destinasi wisata
 - Menanggapi rekomendasi pengguna
 - Melakukan manajemen pengguna
 - Kontroler komentar negatif

Kebutuhan fungsional sistem digambarkan dalam diagram kasus penggunaan seperti pada Gambar 3.2 dibawah ini.



Gambar 3.2 Gambar Kasus Penggunaan

Penjelasan mengenai kasus penggunaan sistem monitoring ini terdapat pada Tabel 3.1

Tabel 3.1 Deskripsi Kasus Penggunaan

No	Kode	Nama	Aktor
1	UC-01	Melakukan registrasi	Pengguna
2	UC-02	Melakukan filter kategori pada menu Home	Pengguna
3	UC-03	Melihat list hasil filter kategori	Pengguna

No	Kode	Nama	Aktor
4	UC-04	Melihat destinasi wisata secara keseluruhan dalam bentuk list	Pengguna
5	UC-05	Melihat destinasi wisata secara keseluruhan dalam bentuk marker pada peta	Pengguna
6	UC-06	Melihat detail informasi wisata	Pengguna
7	UC-07	Memberikan review terkait wisata yang dipilih	Pengguna
8	UC-08	Memberikan informasi terbaru terkait destinasi wisata yang sudah ada kepada admin	Pengguna
9	UC-09	Memberikan rekomendasi wisata baru kepada administrator	Pengguna
10	UC-10	Melihat profile	Pengguna
11	UC-11	Mengedit profile	Pengguna
12	UC-12	Melakukan unggah foto untuk galeri spot wisata	Pengguna
13	UC-13	Mengelola konten destinasi wisata	Administrator
14	UC-14	Menanggapi rekomendasi pengguna	Administrator
15	UC-15	Melakukan manajemen pengguna	Administrator
16	UC-16	Kontroler komentar negatif	Administrator

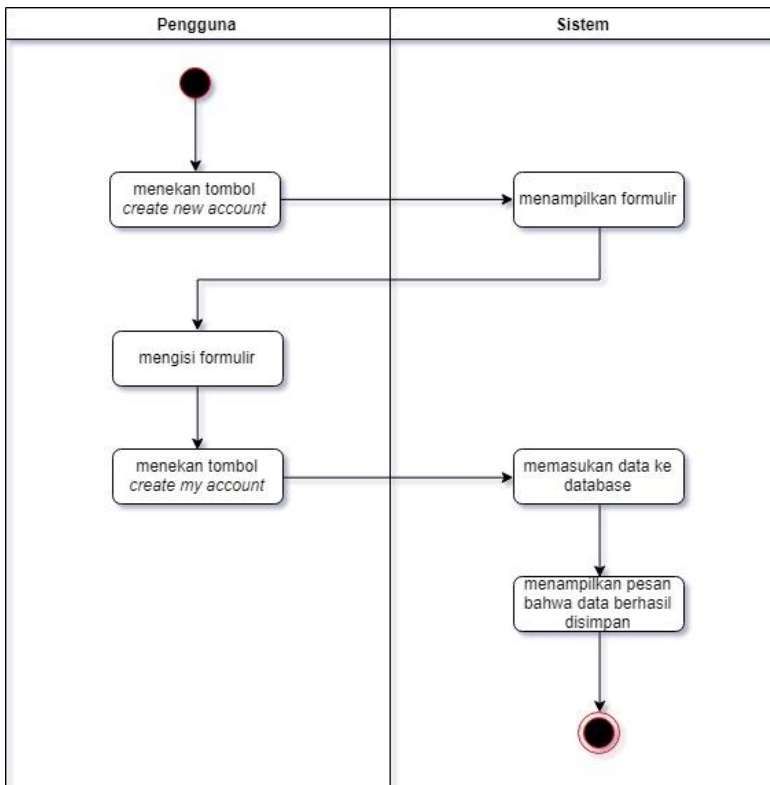
3.1.2.2.1. Deskripsi Kasus Penggunaan UC-01

Kasus penggunaan kode UC-01 merupakan kasus penggunaan melakukan registrasi bagi pengguna baru. Detail alur kasus melakukan registrasi dijelaskan pada Tabel 3.2 beserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.3

Tabel 3.2 Rincian Alur Kasus Penggunaan UC-01

Nama Use Case	Melakukan Registrasi
Nomor	UC-01
Aktor	Pengguna
Kondisi Awal	Data baru belum masuk ke database

Kondisi Akhir	Data baru tersimpan didatabase
Alur Normal	Pengguna menekan tombol <i>create new account</i>
	Sistem menampilkan <i>form</i> berupa <i>full name</i> , <i>username</i> , <i>password</i> dan <i>email</i>
	Pengguna mengisi form input data
	Pengguna menekan tombol <i>create my account</i>
	Sistem memasukan data ke database
	Sistem menampilkan pesan bahwa data berhasil disimpan



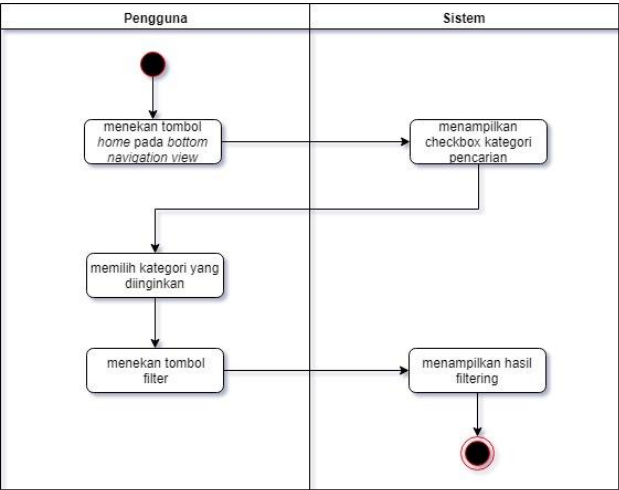
Gambar 3.3 Diagram Aktivitas Kasus Penggunaan UC-01

3.1.2.2.2. Deskripsi Kasus Penggunaan UC-02

Kasus penggunaan kode UC-02 merupakan kasus penggunaan melakukan *filter* kategori pada menu *Home*. Detail alur kasus melakukan *filter* kategori dijelaskan pada Tabel 3.3 beserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.4.

Tabel 3.3 Rincian Alur Kasus Penggunaan UC-02

Nama Use Case	Melakukan filter kategori pada menu Home
Nomor	UC-02
Aktor	Pengguna
Kondisi Awal	Sistem menampilkan kategori pencarian
Kondisi Akhir	Sistem mefilter sesuai kategori yang dipilih
Alur Normal	Pengguna menekan tombol home pada bagian tengah <i>bottom navigation view</i>
	Sistem menampilkan <i>checkbox</i> kategori pencarian
	Pengguna memilih kriteria yang diinginkan
	Pengguna menekan tombol <i>filter</i>
	Sistem menampilkan hasil <i>filter</i>



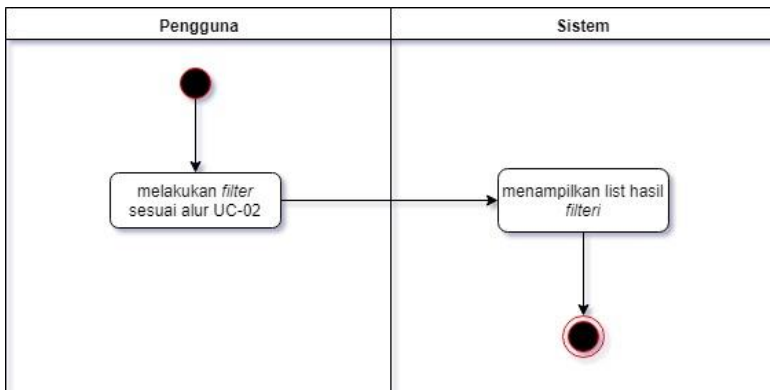
Gambar 3.4 Diagram Aktivitas Kasus Penggunaan UC-02

3.1.2.2.3. Deskripsi Kasus Penggunaan UC-03

Kasus penggunaan kode UC-03 merupakan kasus penggunaan dalam melihat *list* hasil *filter* kategori. Detail alur kasus melihat melihat list hasil *filter* kategori dijelaskan pada Tabel 3.4 beserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.5.

Tabel 3.4 Rincian Alur Kasus Penggunaan UC-03

Nama Use Case	Melihat list hasil filter kategori
Nomor	UC-03
Aktor	Pengguna
Kondisi Awal	Sistem tidak menampilkan hasil filtering
Kondisi Akhir	Sistem menampilkan hasil filtering dalam bentuk card
Alur Normal	Pengguna melakukan <i>filter</i> kategori terlebih dahulu sesuai alur UC-02
	Sistem menampilkan <i>list</i> hasil <i>filter</i>



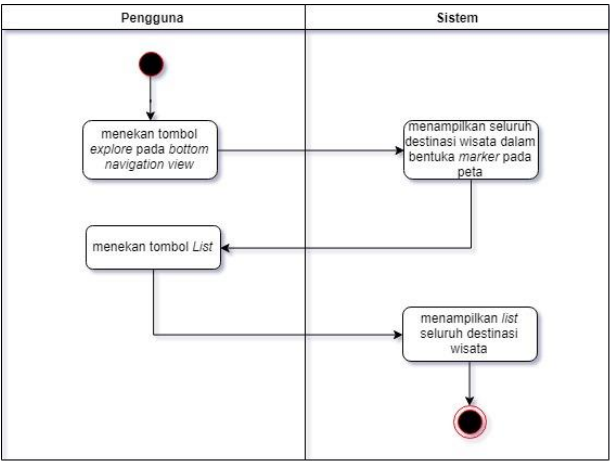
Gambar 3.5 Diagram Aktivitas Kasus Penggunaan UC-03

3.1.2.2.4. Deskripsi Kasus Penggunaan UC-04

Kasus penggunaan kode UC-04 merupakan kasus penggunaan dalam melihat destinasi wisata secara keseluruhan dalam bentuk *list*. Detail alur kasus melihat destinasi wisata secara keseluruhan dalam bentuk *list* dijelaskan pada Tabel 3.5 beserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.6.

Tabel 3.5 Rincian Alur Kasus Penggunaan UC-04

Nama Use Case	Melihat destinasi wisata secara keseluruhan dalam bentuk list
Nomor	UC-04
Aktor	Pengguna
Kondisi Awal	Sistem tidak menampilkan destinasi wisata
Kondisi Akhir	Sistem menampilkan seluruh destinasi wisata dalam bentuk list
Alur Normal	Pengguna menekan tombol <i>Explore</i> pada <i>bottom navigation view</i>
	Sistem menampilkan seluruh destinasi wisata dalam bentuk <i>marker</i> pada peta
	Pengguna menekan tombol <i>List</i>
	Sistem menampilkan <i>list</i> seluruh destinasi wisata



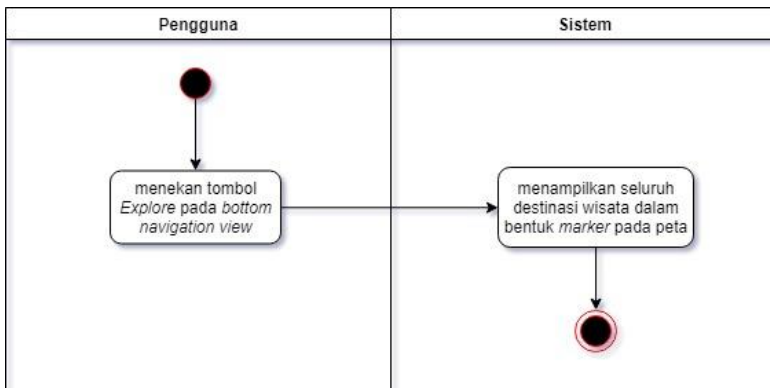
Gambar 3.6 Diagram Aktivitas Kasus Penggunaan UC-04

3.1.2.2.5. Deskripsi Kasus Penggunaan UC-05

Kasus penggunaan kode UC-05 merupakan kasus penggunaan dalam melihat destinasi wisata secara keseluruhan dalam bentuk *marker* pada peta. Detail alur kasus melihat destinasi wisata secara keseluruhan dalam bentuk *marker* pada peta dijelaskan pada Tabel 3.6 berserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.7.

Tabel 3.6 Rincian Alur Kasus Penggunaan UC-05

Nama Use Case	Melihat destinasi wisata secara keseluruhan dalam bentuk marker pada peta
Nomor	UC-05
Aktor	Pengguna
Kondisi Awal	Sistem tidak menampilkan destinasi wisata
Kondisi Akhir	Sistem menampilkan seluruh destinasi wisata dalam bentuk <i>marker</i> pada peta
Alur Normal	Pengguna menekan tombol <i>Explore</i> pada <i>bottom navigation view</i>
	Sistem menampilkan seluruh destinasi wisata dalam bentuk <i>marker</i> pada peta



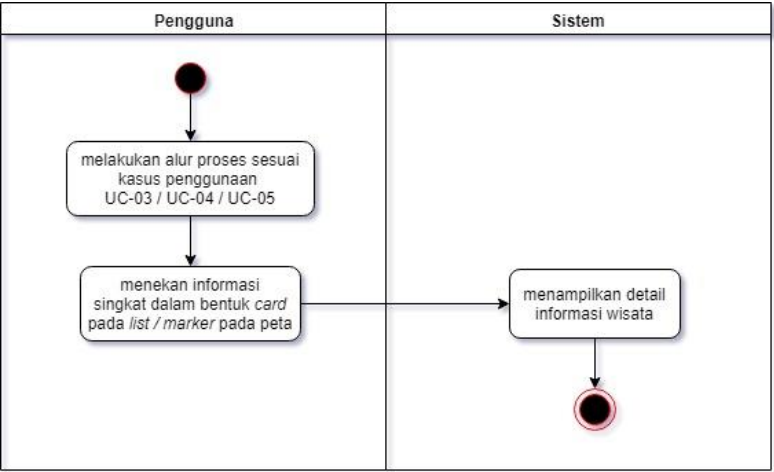
Gambar 3.7 Diagram Aktivitas Kasus Penggunaan UC-05

3.1.2.2.6. Deskripsi Kasus Penggunaan UC-06

Kasus penggunaan kode UC-06 merupakan kasus penggunaan dalam melihat detail informasi wisata yang dipilih setelah melakukan alur proses sebelumnya. Detail alur kasus melihat detail informasi wisata dijelaskan pada Tabel 3.7 beserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.8.

Tabel 3.7 Rincian Alur Kasus Penggunaan UC-06

Nama <i>Use Case</i>	Melihat detail informasi wisata
Nomor	UC-06
Aktor	Pengguna
Kondisi Awal	Sistem menampilkan informasi singkat dalam bentuk <i>card</i> pada <i>list / marker</i> pada peta
Kondisi Akhir	Sistem menampilkan detail informasi
Alur Normal	Pengguna melakukan alur proses sesuai kasus penggunaan UC-03 / UC-04 / UC-05
	Pengguna menekan informasi singkat dalam bentuk <i>card</i> pada <i>list / marker</i> pada peta
	Sistem menampilkan detail informasi wisata



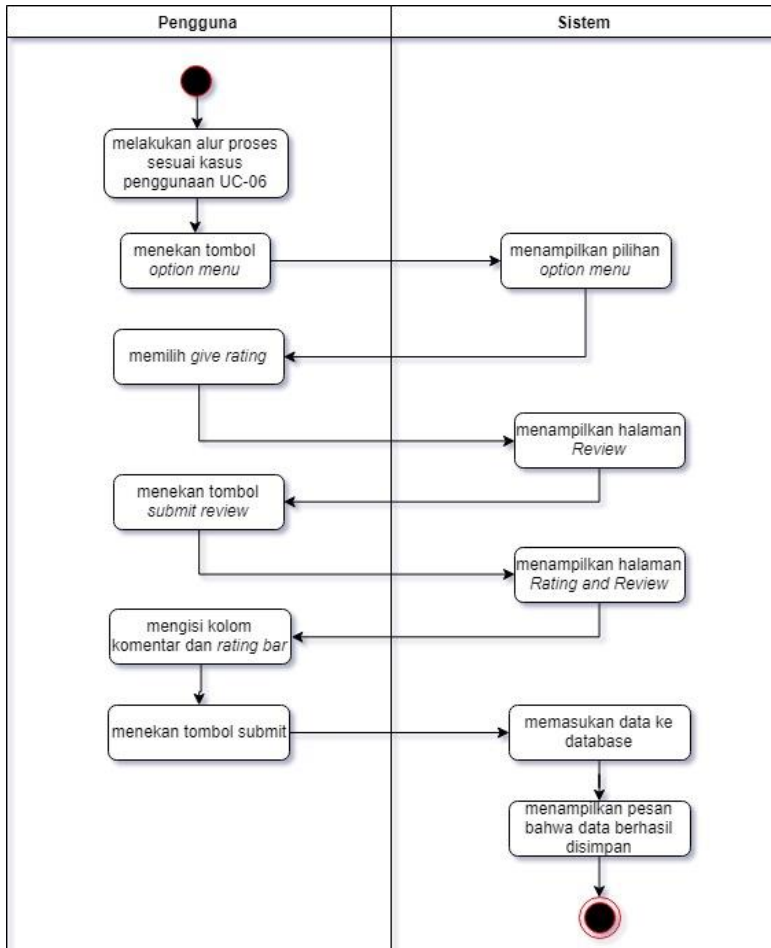
Gambar 3.8 Diagram Aktivitas Kasus Penggunaan UC-06

3.1.2.2.7. Deskripsi Kasus Penggunaan UC-07

Kasus penggunaan kode UC-07 merupakan kasus penggunaan dalam memberikan *review* terkait destinasi wisata yang dipilih. Detail alur kasus memberikan *review* terkait destinasi wisata yang dipilih dijelaskan pada Tabel 3.8 beserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.9.

Tabel 3.8 Rincian Alur Kasus Penggunaan UC-07

Nama Use Case	Memberikan review terkait destinasi wisata yang dipilih
Nomor	UC-07
Aktor	Pengguna
Kondisi Awal	Review wisata belum masuk ke database
Kondisi Akhir	Review tersimpan didatabase
Alur Normal	Pengguna melakukan alur proses sesuai kasus penggunaan UC-06
	Pengguna menekan tombol <i>option menu</i> pada bagian pojok atas kanan
	Sistem menampilkan pilihan <i>option menu</i>
	Pengguna memilih <i>give rating</i>
	Sistem menampilkan halaman <i>Review</i>
	Pengguna menekan tombol <i>Submit Review</i>
	Sistem menampilkan halaman <i>Rating and Review</i>
	Pengguna mengisi kolom komentar dan <i>rating bar</i>
	Pengguna menekan tombol <i>Submit</i>
	Sistem memasukan data ke database
	Sistem menampilkan pesan bahwa data berhasil disimpan



Gambar 3.9 Diagram Aktivitas Kasus Penggunaan UC-07

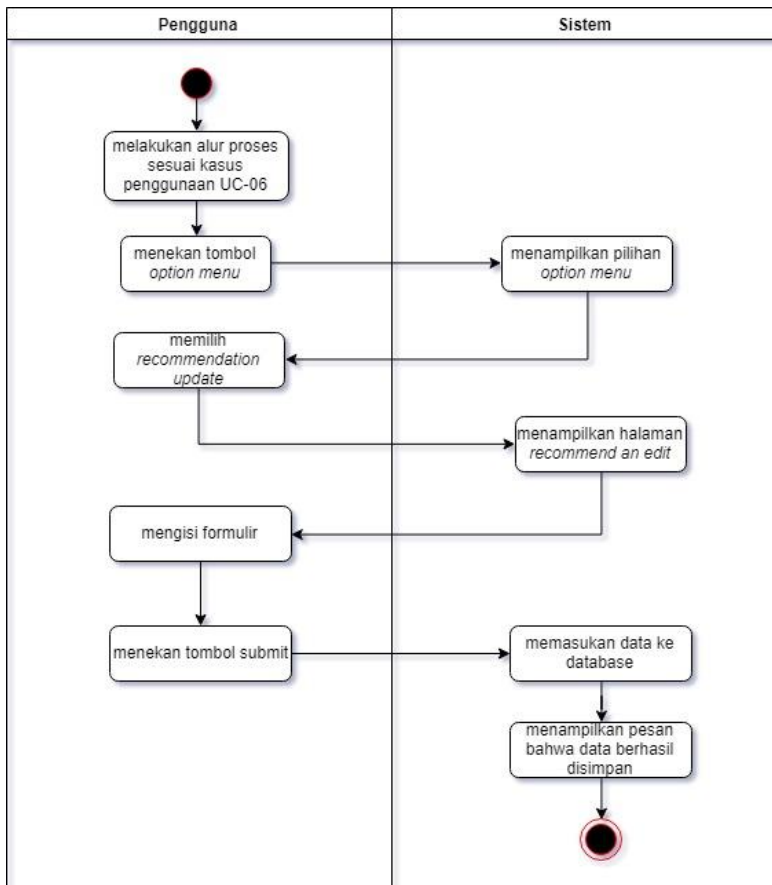
3.1.2.2.8. Deskripsi Kasus Penggunaan UC-08

Kasus penggunaan kode UC-08 merupakan kasus penggunaan dalam pengguna memberikan rekomendasi wisata kepada administrator yang tersedia pada layanan *profile*. Detail alur kasus memberi rekomendasi wisata kepada administrator dijelaskan pada

Tabel 3.9 beserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.10.

Tabel 3.9 Rincian Alur Kasus Penggunaan UC-08

Nama Use Case	Memberikan informasi terbaru terkait destinasi wisata yang sudah ada kepada administrator
Nomor	UC-08
Aktor	Pengguna
Kondisi Awal	Informasi wisata belum <i>terupdate</i>
Kondisi Akhir	Data informasi wisata terbaru masuk ke database
Alur Normal	Pengguna melakukan alur proses sesuai kasus penggunaan UC-06
	Pengguna menekan tombol <i>option menu</i> pada bagian pojok atas kanan
	Sistem menampilkan pilihan <i>option menu</i>
	Pengguna memilih <i>recommendation update</i>
	Sistem menampilkan halaman <i>recommend an edit</i>
	Pengguna mengisi formulir
	Pengguna menekan tombol <i>Submit</i>
	Sistem memasukan data ke database
	Sistem menampilkan pesan bahwa data berhasil disimpan



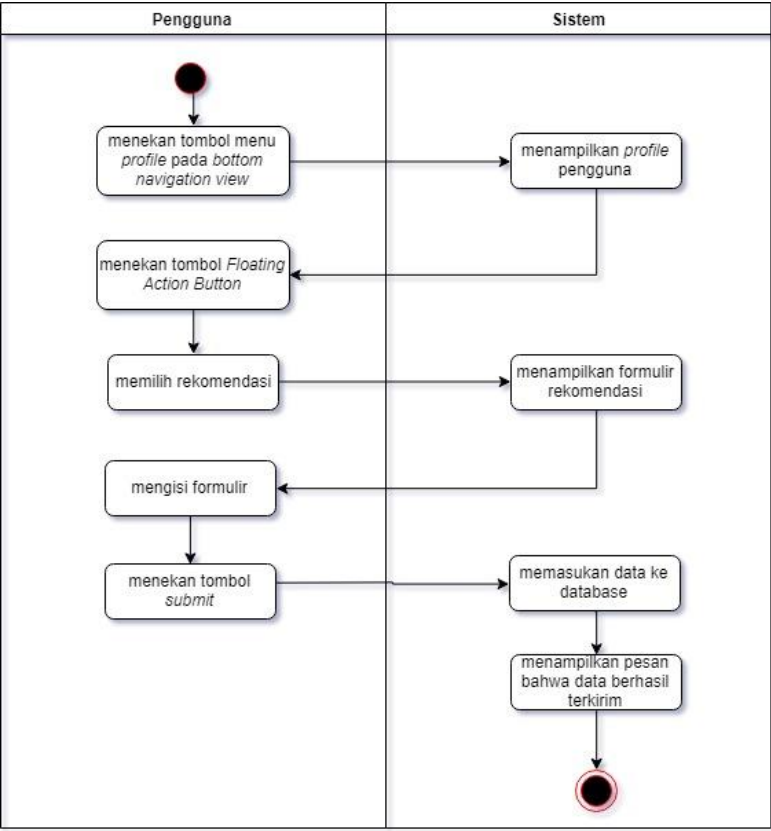
Gambar 3.10 Diagram Aktivitas Kasus Penggunaan UC-08

3.1.2.2.9. Deskripsi Kasus Penggunaan UC-09

Kasus penggunaan kode UC-09 merupakan kasus penggunaan dalam pengguna memberikan rekomendasi wisata kepada administrator yang tersedia pada layanan *profile*. Detail alur kasus memberi rekomendasi wisata kepada administrator dijelaskan pada Tabel 3.10 beserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.11.

Tabel 3.10 Rincian Alur Kasus Penggunaan UC-09

Nama Use Case	Memberikan rekomendasi wisata kepada administrator
Nomor	UC-09
Aktor	Pengguna
Kondisi Awal	Data rekomendasi belum masuk ke database
Kondisi Akhir	Data rekomendasi masuk ke database
Alur Normal	Pengguna menekan tombol menu <i>profile</i> pada <i>bottom navigation view</i>
	Sistem menampilkan <i>profile</i> pengguna
	Pengguna menekan tombol <i>Floating Action Button</i>
	Pengguna menekan memilih rekomendasi
	Sistem menampilkan formulir rekomendasi
	Pengguna mengisi formulir
	Pengguna menekan tombol <i>submit</i>
	Sistem memasukan data ke database
	Sistem menampilkan pesan bahwa data berhasil diupload



Gambar 3.11 Diagram Aktivitas Kasus Penggunaan UC-09

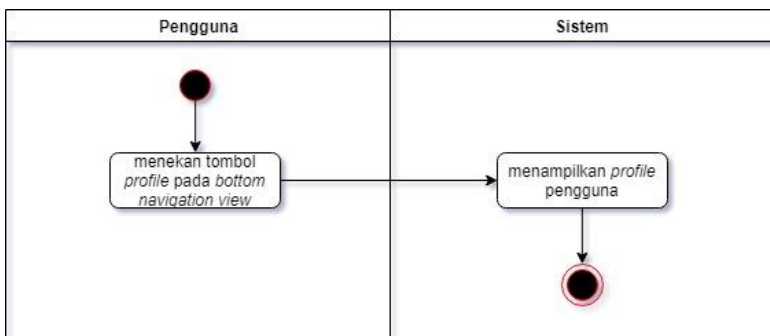
3.1.2.2.10. Deskripsi Kasus Penggunaan UC-10

Kasus penggunaan kode UC-10 merupakan kasus penggunaan dalam melihat *profile* pengguna. Detail alur kasus melihat *profile* dijelaskan pada Tabel 3.11 beserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.12.

Tabel 3.11 Rincian Alur Kasus Penggunaan UC-10

Nama Use Case	Melihat profile
Nomor	UC-10

Aktor	Pengguna
Kondisi Awal	Sistem tidak menampilkan <i>profile</i>
Kondisi Akhir	Sistem menampilkan <i>profile</i>
Alur Normal	Pengguna menekan tombol <i>profile</i> pada <i>bottom navigation view</i>
	Sistem menampilkan <i>profile</i> pengguna



Gambar 3.12 Diagram Aktivitas Kasus Penggunaan UC-10

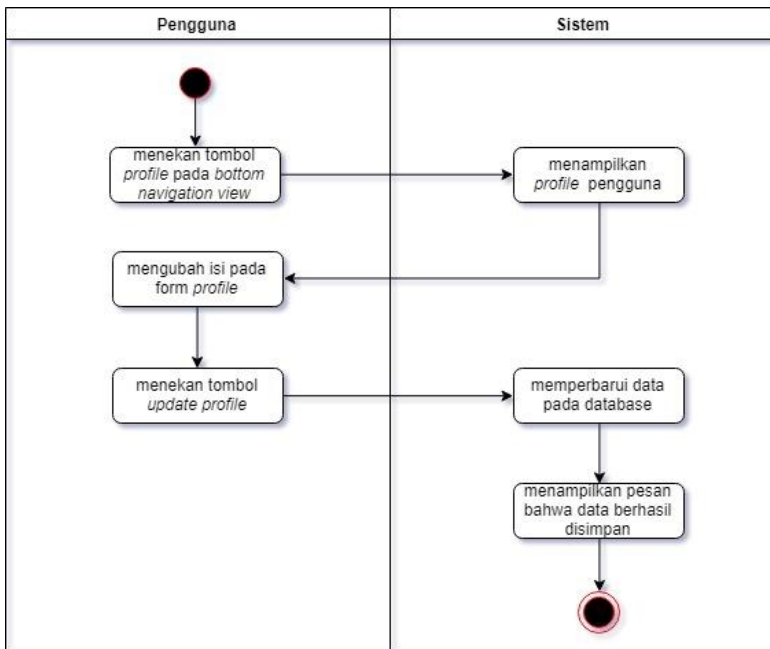
3.1.2.2.11. Deskripsi Kasus Penggunaan UC-11

Kasus penggunaan kode UC-11 merupakan kasus penggunaan dalam mengedit *profile* meliputi penggantian foto profil, nama, *username*, no hp, email dan *password*. Detail alur kasus mengedit *profile* dijelaskan pada Tabel 3.12 berserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.13.

Tabel 3.12 Rincian Alur Kasus Penggunaan UC-11

Nama Use Case	Mengedit profile
Nomor	UC-11
Aktor	Pengguna
Kondisi Awal	Data belum <i>terupdate</i>

Kondisi Akhir	<i>Update data baru tersimpan di database</i>
Alur Normal	Pengguna menekan tombol menu <i>profile</i> pada <i>bottom navigation view</i>
	Sistem menampilkan <i>profile</i> pengguna
	Pengguna mengubah isi pada formulir <i>profile</i>
	Pengguna menekan tombol <i>update profile</i>
	Sistem memperbarui data pada database
	Sistem menampilkan pesan bahwa data berhasil disimpan



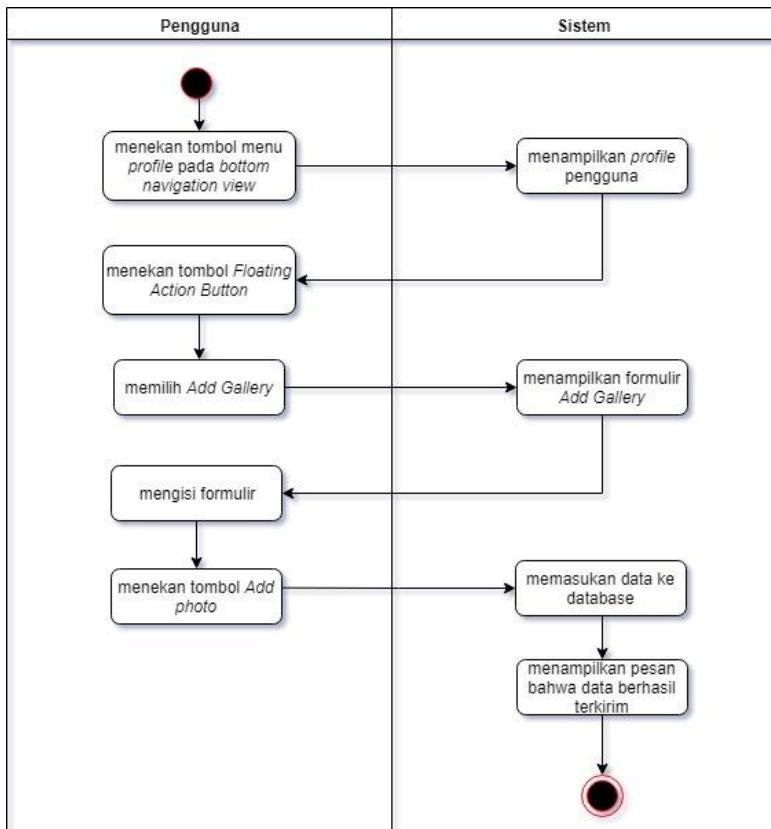
Gambar 3.13 Diagram Aktivitas Kasus Penggunaan UC-11

3.1.2.2.12. Deskripsi Kasus Penggunaan UC-11

Kasus penggunaan kode UC-12 merupakan kasus penggunaan dalam melakukan unggah foto untuk galeri spot wisata. Detail alur kasus melakukan unggah foto untuk galeri spot wisata dijelaskan pada Tabel 3.13 beserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.14

Tabel 3.13 Rincian Alur Kasus Penggunaan UC-12

Nama Use Case	Melakukan unggah foto untuk galeri spot wisata
Nomor	UC-12
Aktor	Pengguna
Kondisi Awal	Data belum masuk ke database
Kondisi Akhir	Data masuk ke database
Alur Normal	Pengguna menekan tombol menu <i>profile</i> pada <i>bottom navigation view</i>
	Sistem menampilkan <i>profile</i> pengguna
	Pengguna menekan tombol <i>Floating Action Button</i>
	Pengguna memilih <i>Add Gallery</i>
	Sistem menampilkan halaman <i>Add Gallery</i>
	Pengguna mengisi formulir
	Pengguna menekan tombol <i>Add photo</i>
	Sistem memasukan data ke database
	Sistem menampilkan pesan bahwa data berhasil diupload



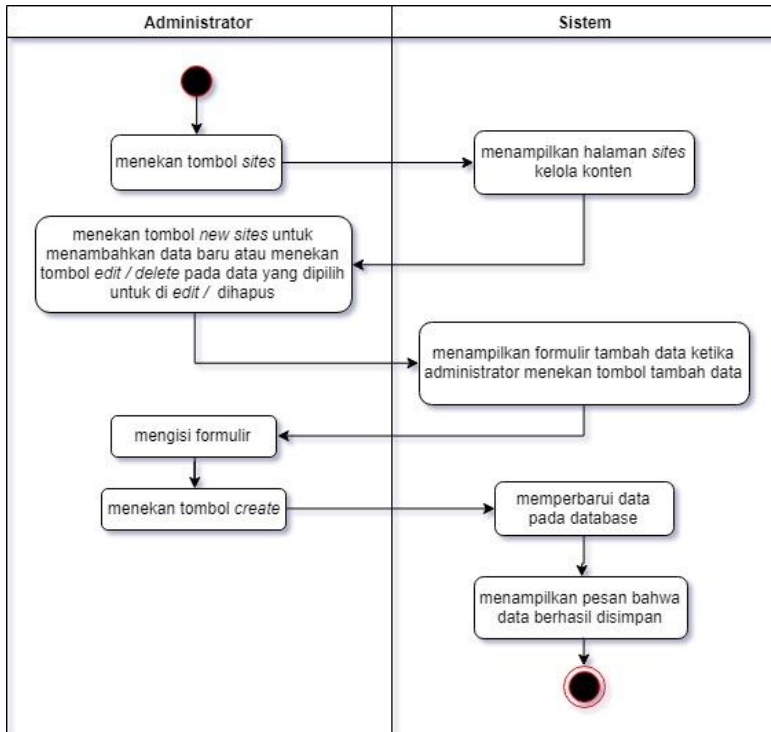
Gambar 3.14 Diagram Aktivitas Kasus Penggunaan UC-12

3.1.2.2.13. Deskripsi Kasus Penggunaan UC-13

Kasus penggunaan kode UC-13 merupakan kasus penggunaan dalam administrator mengelola konten destinasi wisata meliputi tambah, *edit* dan hapus. Detail alur kasus mengelola konten destinasi wisata dijelaskan pada Tabel 3.14 berserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.15

Tabel 3.14 Rincian Alur Kasus Penggunaan UC-13

Nama Use Case	Mengelola konten destinasi wisata
Nomor	UC-13
Aktor	Administrator
Kondisi Awal	Sistem belum menampilkan halaman kelola konten
Kondisi Akhir	Sistem berhasil melakukan perubahan data sesuai perintah yang dijalankan
Alur Normal	Administrator menekan tombol <i>sites</i>
	Sistem menampilkan halaman <i>sites</i> kelola konten
	Administrator menekan tombol <i>new sites</i> untuk menambahkan data baru atau menekan tombol <i>edit / delete</i> pada data yang dipilih untuk di <i>edit / dihapus</i>
	Sistem menampilkan formulir tambah data ketika administrator menekan tombol tambah data
	Administrator mengisi formulir
	Administrator menekan tombol <i>create</i>
	Sistem memperbarui data pada database
	Sistem menampilkan pesan bahwa data berhasil disimpan



Gambar 3.15 Diagram Aktivitas Kasus Penggunaan UC-13

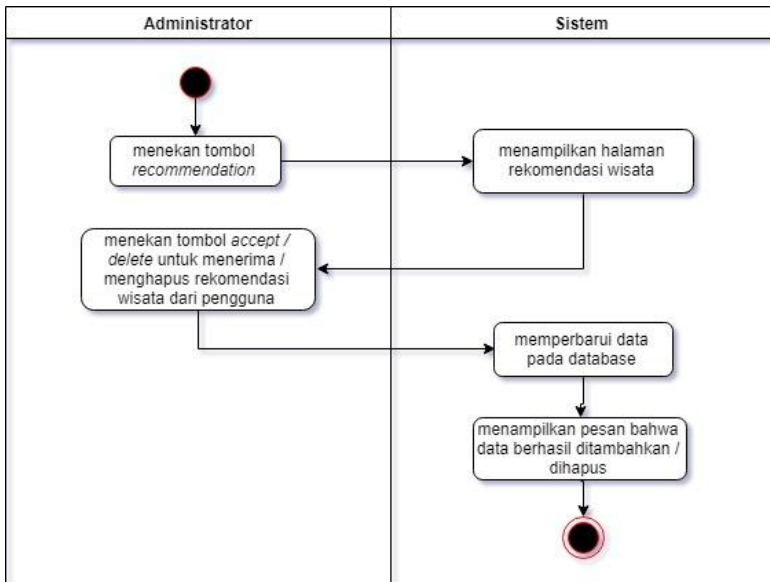
3.1.2.2.14. Deskripsi Kasus Penggunaan UC-14

Kasus penggunaan kode UC-14 merupakan kasus penggunaan dalam administrator menanggapi rekomendasi wisata dari pengguna baik itu wisata baru ataupun *update* informasi dari wisata yang telah ada sebelumnya. Detail alur kasus menanggapi rekomendasi pengguna dijelaskan pada Tabel 3.15. beserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.16

Tabel 3.15 Rincian Alur Kasus Penggunaan UC-14

Nama Use Case	Menanggapi rekomendasi pengguna
Nomor	UC-14
Aktor	Administrator

Kondisi Awal	Sistem belum menampilkan halaman rekomendasi wisata
Kondisi Akhir	Sistem berhasil menambahkan data baru sesuai rekomendasi pengguna
Alur Normal	Administrator menekan tombol <i>recommendation</i>
	Sistem menampilkan halaman rekomendasi wisata
	Administrator menekan tombol <i>accept / delete</i> untuk menerima atau menghapus rekomendasi wisata dari pengguna
	Sistem memperbarui data pada database
	Sistem menampilkan pesan bahwa data berhasil ditambahkan / dihapus



Gambar 3.16 Diagram Aktivitas Kasus Penggunaan UC-14

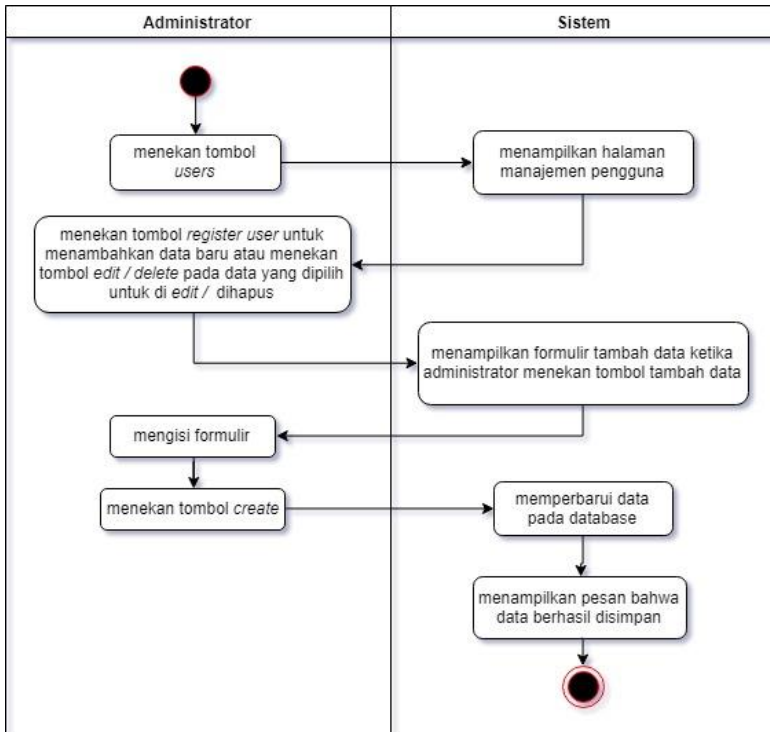
3.1.2.2.15. Deskripsi Kasus Penggunaan UC-15

Kasus penggunaan kode UC-15 merupakan kasus penggunaan dalam administrator melakukan manajemen pengguna meliputi tambah, ubah dan hapus. Detail alur kasus manajemen pengguna

dijelaskan pada Tabel 3.16 berserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.17.

Tabel 3.16 Rincian Alur Kasus Penggunaan UC-15

Nama Use Case	Melakukan manajemen pengguna
Nomor	UC-15
Aktor	Administrator
Kondisi Awal	Sistem belum menampilkan halaman manajemen pengguna
Kondisi Akhir	Sistem berhasil melakukan perubahan data sesuai perintah yang dijalankan
Alur Normal	Administrator menekan tombol <i>users</i>
	Sistem menampilkan halaman manajemen pengguna
	Administrator menekan tombol <i>register user</i> untuk menambahkan data baru atau menekan tombol <i>edit / delete</i> pada data yang dipilih untuk di <i>edit / dihapus</i>
	Sistem menampilkan formulir tambah <i>user</i> ketika administrator menekan tombol <i>register user</i>
	Administrator mengisi formulir
	Administrator menekan tombol <i>create</i>
	Sistem menambah data pada database
	Sistem menampilkan pesan bahwa data berhasil disimpan



Gambar 3.17 Diagram Aktivitas Kasus Penggunaan UC-15

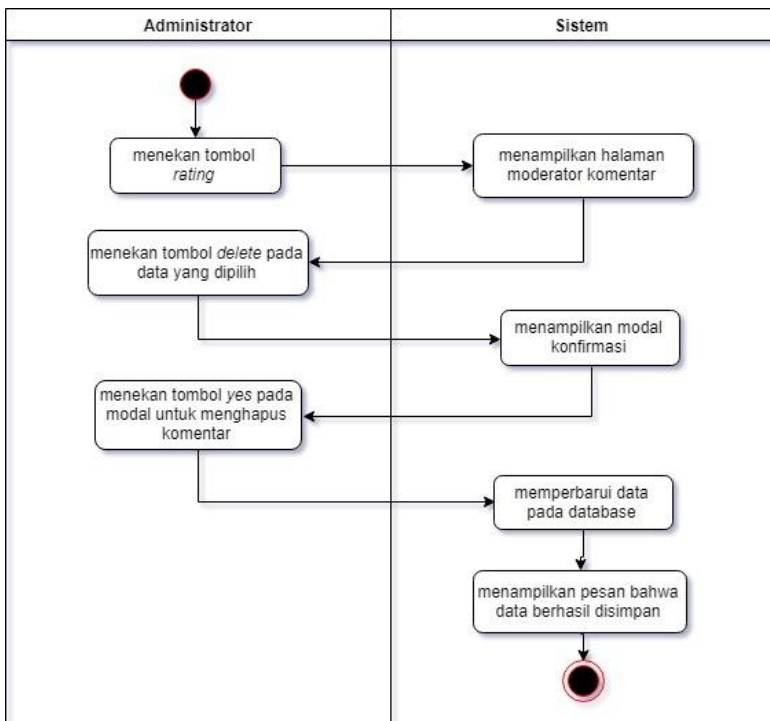
3.1.2.2.16. Deskripsi Kasus Penggunaan UC-16

Kasus penggunaan kode UC-16 merupakan kasus penggunaan dalam administrator mengawasi komentar negative dari pengguna, administrator bertugas sebagai moderator. Detail alur kasus kontrol komentar negatif dijelaskan pada Tabel 3.17 berserta diagram aktivitas kasus yang dijelaskan pada Gambar 3.18.

Tabel 3.17 Rincian Alur Kasus Penggunaan UC-16

Nama Use Case	Kontroler komentar negatif
Nomor	UC-16
Aktor	Administrator

Kondisi Awal	Sistem belum menampilkan halaman moderator komentar
Kondisi Akhir	Sistem berhasil melakukan penghapusan komentar negatif
Alur Normal	Administrator menekan tombol <i>rating</i>
	Sistem menampilkan halaman moderator komentar
	Administrator menekan tombol <i>delete</i> pada data yang dipilih
	Sistem menampilkan modal konfirmasi
	Administrator menekan tombol <i>yes</i> pada modal untuk menghapus komentar
	Sistem memperbarui data pada database
	Sistem menampilkan pesan bahwa data berhasil dihapus



Gambar 3.18 Diagram Aktivitas Kasus Penggunaan UC-16

3.1.2.3. Kebutuhan Non Fungsional

Kebutuhan sistem yang diperlukan dalam membangun sistem *monitoring* ini terdiri dari kebutuhan fungsional dan non-fungsional. Kebutuhan non fungsional yang diperlukan dijelaskan pada Tabel 3.18:

Tabel 3.18 Kebutuhan Non Fungsional

No	Parameter	Keterangan
1	Ketersediaan	Aplikasi harus dapat berjalan pada sistem operasi yang sesuai dengan platform yang disebutkan
2	<i>Portability</i>	Aplikasi mudah untuk dioperasikan pada semua perangkat digital dengan mengakses <i>smartphone android</i> untuk pengguna dan <i>website</i> untuk administratorisrator
3	Bahasa	Bahasa yang digunakan untuk antarmuka menggunakan Bahasa Inggris

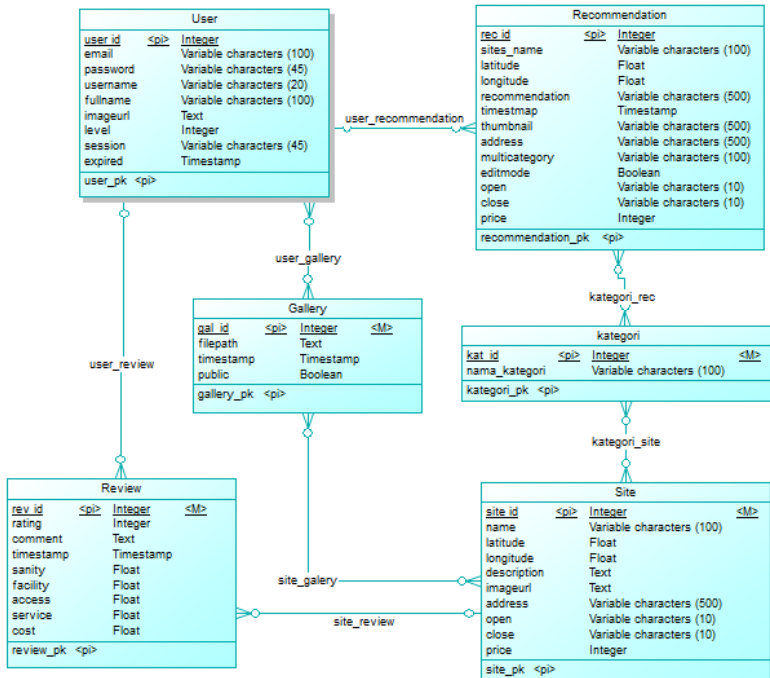
3.2. Perancangan Sistem

Tahap ini berupa perancangan basis data sistem, analisis proses sistem, analisis antarmuka aplikasi panduan wisata. Pembahasan lebih lanjut akan dibahas sebagai berikut.

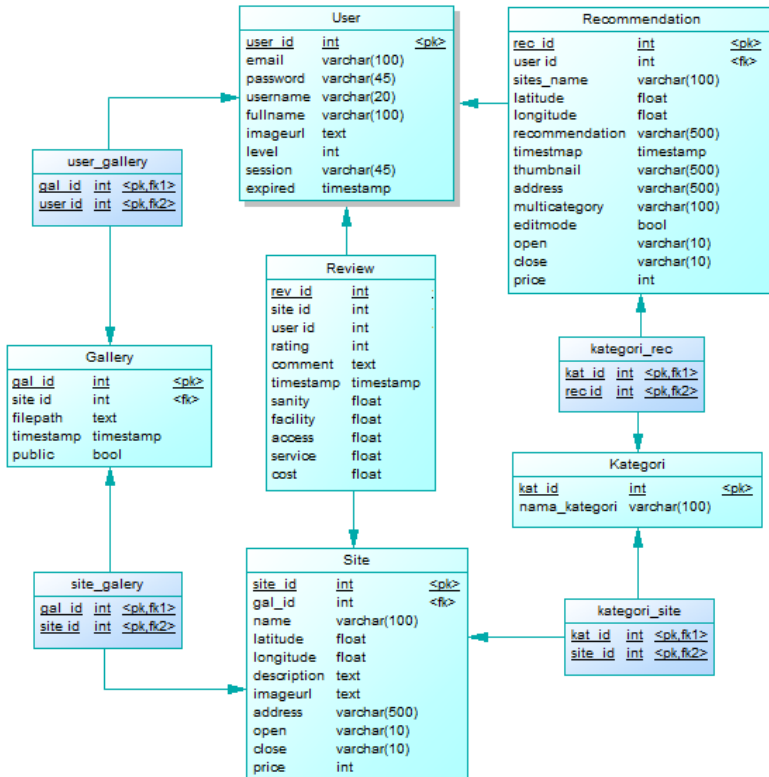
3.2.1. Perancangan Basis Data

Pada subbab ini akan membahas bagaimana rancangan basis data yang digunakan pada aplikasi panduan wisata. Basis data yang digunakan adalah MySQL. MySQL digunakan untuk menyimpan data user, recommendation, review, site dan kategori .

Conceptual Data Model (CDM) dan *Physical Data Model* (PDM) dari basis data dapat dilihat pada Gambar 3.19 dan Gambar 3.20



Gambar 3.19 CDM (Conceptual Data Model) pada web service



Gambar 3.20 PDM (Physical Data Model) pada web service

3.2.1.1. Rancangan Tabel User

Tabel user adalah tabel yang digunakan sebagai penyimpanan data *user*. Detail Atribut Tabel *User* dijelaskan pada Tabel 3.19

Tabel 3.19 Atribut Tabel Data User

Nama Kolom	Type Data	Keterangan
User_id	<i>integer</i>	<i>Primary Key</i> pada tabel <i>User</i>
email	<i>Varchar(100)</i>	Email dari <i>user</i>
password	<i>Varchar(45)</i>	Password dari <i>user</i>
username	<i>Varchar(20)</i>	Username dari <i>user</i>

Nama Kolom	Tipe Data	Keterangan
fullname	<i>Varchar(100)</i>	Nama Panjang dari <i>user</i>
imageurl	<i>text</i>	Gambar profil dari <i>user</i>
level	<i>integer</i>	Jenis level <i>user</i>
session	<i>Varchar(45)</i>	Session dari <i>user</i>
expired	<i>timestamp</i>	Expired dari user session

3.2.1.2. Rancangan Tabel Recommendation

Tabel *Recommendation* adalah tabel yang digunakan untuk menyimpan data rekomendasi dari pengguna. Detail Atribut Tabel *Recommendation* dijelaskan pada Tabel 3.20.

Tabel 3.20 Atribut Tabel Data Recommendation

Nama Kolom	Tipe Data	Keterangan
rec_id	<i>integer</i>	<i>Primary Key</i> pada tabel <i>Recommendation</i>
User_id	<i>integer</i>	<i>Foreign Key</i> dari tabel <i>user</i>
sites_name	<i>text</i>	Nama dari site
latitude	<i>float</i>	<i>latitude</i> pada Gmaps
longitude	<i>float</i>	<i>Longitude</i> pada Gmaps
recommendation	<i>text</i>	Deskripsi site rekomendasi
timestamp	<i>timestamp</i>	Menyimpan kapan data dibuat
thumbnail	<i>Varchar(500)</i>	Menyimpan gambar thumbnail
address	<i>Varchar(500)</i>	Alamat wisata
multicategory	<i>Varchar(100)</i>	Menyimpan array dari id kategori
editmode	<i>boolean</i>	Flag untuk membedakan rekomendasi dari hasil <i>update info</i> atau <i>new recommendation</i>
open	<i>Varchar(10)</i>	Jam buka wisata
close	<i>Varchar(10)</i>	Jam tutup wisata
price	<i>integer</i>	Harga tiket masuk wisata

3.2.1.3. Rancangan Tabel Site

Tabel *Site* adalah tabel yang digunakan untuk menyimpan data destinasi wisata. Detail Atribut Tabel *Site* dijelaskan pada Tabel 3.21.

Tabel 3.21 Atribut Tabel Site

Nama Kolom	Tipe Data	Keterangan
site_id	<i>integer</i>	<i>Primary Key</i> pada tabel Jenis
gal_id	<i>integer</i>	<i>Foreign Key</i> dari tabel <i>gallery</i>
name	<i>varchar(32)</i>	Nama dari site
latitude	<i>float</i>	<i>Latitude</i> pada Gmaps
longitude	<i>float</i>	<i>Longitude</i> pada Gmaps
description	<i>text</i>	Deskripsi dari site
imageurl	<i>text</i>	Gambar dari site
Address	<i>Varchar(500)</i>	Alamat wisata
open	<i>Varchar(10)</i>	Jam buka wisata
close	<i>Varchar(10)</i>	Jam tutup wisata
price	<i>integer</i>	Harga tiket masuk wisata

3.2.1.4. Rancangan Tabel Review

Tabel *Review* adalah tabel yang digunakan untuk menyimpan hasil review dari pengguna. Detail Atribut Tabel *Review* dijelaskan pada Tabel 3.22.

Tabel 3.22 Atribut Tabel Review

Nama Kolom	Tipe Data	Keterangan
rev_id	<i>long</i>	<i>Primary Key</i> pada tabel Users
Site_id	<i>Integer</i>	<i>Foreign Key</i> dari tabel <i>Site</i>
User_id	<i>integer</i>	<i>Foreign Key</i> dari tabel <i>Review</i>
rating	<i>integer</i>	Rating dari site
comment	<i>text</i>	Comment dari site
timestamp	<i>timestamp</i>	Menyimpan kapan data dibuat
sanity	<i>float</i>	<i>Rating</i> kebersihan
facility	<i>float</i>	<i>Rating</i> fasilitas

Nama Kolom	Tipe Data	Keterangan
access	<i>float</i>	<i>Rating akses jalan</i>
service	<i>float</i>	<i>Rating pelayanan</i>
cost	<i>float</i>	<i>Rating biaya</i>

3.2.1.5. Rancangan Tabel Kategori

Tabel kategori adalah tabel yang digunakan untuk menyimpan macam jenis kategori untuk fitur filtering nantinya.

Detail Atribut Tabel Kategori dijelaskan pada Tabel 3.23.

Tabel 3.23 Atribut Tabel Kategori

Nama Kolom	Tipe Data	Keterangan
kat_id	<i>integer</i>	<i>Primary Key</i> pada tabel Kategori
nama_kategori	<i>varchar(100)</i>	Nama dari kategori

3.2.1.6. Rancangan Tabel Kategori site

Tabel kategori_site adalah tabel yang digunakan sebagai penghubung antara tabel kategori dan tabel site.

Detail Atribut Tabel kategori_site dijelaskan pada Tabel 3.24 Atribut Tabel Kategori_site.

Tabel 3.24 Atribut Tabel Kategori_site

Nama Kolom	Tipe Data	Keterangan
kat_id	<i>integer</i>	<i>Foreign Key</i> pada tabel Kategori_site
site_id	<i>integer</i>	<i>Foreign Key</i> dari tabel Kategori_site

3.2.1.7. Rancangan Tabel Gallery

Tabel *gallery* adalah tabel yang digunakan untuk menyimpan foto yang diunggah pada *profile* maupun *site*.

Detail Atribut Tabel *Gallery* dijelaskan pada Tabel 3.25.

Tabel 3.25 Atribut Tabel Gallery

Nama Kolom	Tipe Data	Keterangan
gal_id	<i>integer</i>	<i>Primary Key</i> pada tabel <i>Gallery</i>
site_id	<i>integer</i>	<i>Foreign Key</i> pada tabel <i>Site</i>

Nama Kolom	Tipe Data	Keterangan
filepath	<i>text</i>	
timestamp	<i>timestamp</i>	
public	<i>bool</i>	

3.2.2. Perancangan Web Service

Pada subbab ini akan membahas desain *web service* yang akan diimplementasikan pada tugas akhir ini. *Web service* yang digunakan adalah RESTful *web service*. Berikut ini adalah fungsi-fungsi yang akan dibangun pada *web service*.

Perancangan API Endpoint untuk Aplikasi Wisata Magelang dijelaskan pada Tabel 3.26 dibawah ini.

Tabel 3.26 Perancangan API Endpoint

No	Endpoint URI	Method	Deskripsi
Situs Wisata			
1	/controller/site/get.php	GET	Mendapatkan daftar situs wisata
2	/controller/site/create.php	POST	Menambahkan situs wisata
3	/controller/site/delete.php	DELETE	Menghapus situs wisata
Rekomendasi Wisata			
4	/controller/recommendation/get.php	GET	Mendapatkan daftar rekomendasi situs wisata dari pengguna
5	/controller/recommendation/delete.php	DELETE	Menghapus rekomendasi situs wisata dari pengguna
Review Destinasi Wisata			
6	/controller/review/get.php	GET	Mendapatkan daftar review

			dari pengguna
7	/controller/review/delete.php	DELETE	Menghapus review dari pengguna
Pengguna			
8	/controller/user/getalluser.php	GET	Mendapatkan daftar detail profile pengguna
9	/controller/user/register.php	POST	Mengirim detail profile pengguna
10	/controller/user/delete.php	DELETE	Menghapus profile pengguna
11	/controller/user/sessioncheck.php	POST	Mengecek status <i>session</i>

3.2.3. Perancangan Antarmuka

Pada subbab ini akan dibahas dengan terperinci dari rancangan antarmuka sistem. Antarmuka sistem terdiri dari antarmuka pada web administrator dan antarmuka pada aplikasi Android. Berikut ini dibahas antarmuka yang terdapat pada sistem yang dibangun untuk tugas akhir ini.

3.2.3.1. Antarmuka Form Login pada Android

Pada Gambar 3.21, dapat dilihat rancangan antarmuka halaman awal yang berguna untuk login bagi pengguna yang telah memiliki akun, maupun *create new account* untuk registrasi bagi yang belum memiliki akun.

3.2.3.2. Antarmuka Form Register pada Android

Pada Gambar 3.22 dapat dilihat rancangan antarmuka form registrasi yang berguna untuk pengguna baru yang akan membuat akun.

Gambar 3.21 Desain Antar Muka Form Login

Gambar 3.22 Desain Antar Muka Form Register

3.2.3.3. Antarmuka Fitur Filter Kategori pada Android

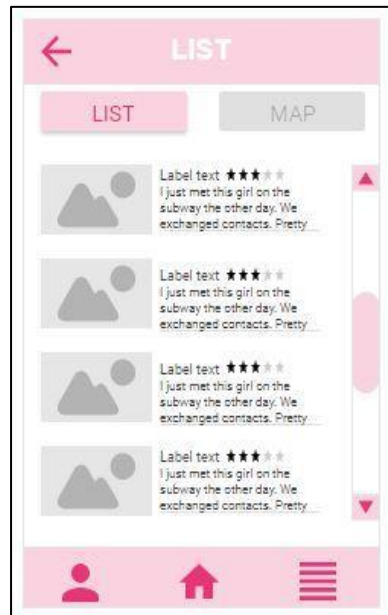
Pada Gambar 3.23 dapat dilihat rancangan antarmuka fitur *filter* kategori yang berguna untuk memfilter destinasi wisata sesuai kriteria yang diinginkan.

3.2.3.4. Antarmuka List Destinasi Wisata pada Android

Pada Gambar 3.24 dapat dilihat rancangan antarmuka *list* destinasi wisata yang berisi informasi singkat wisata dari hasil *filter* yang telah dilakukan sebelumnya.



Gambar 3.23 Desain Antarmuka Fitur Filter Kategori



Gambar 3.24 Desain Antarmuka List hasil Filter Kategori

3.2.3.5. Antarmuka Marker Peta Destinasi Wisata pada Android

Pada Gambar 3.25 dapat dilihat rancangan antarmuka *marker* destinasi wisata pada peta.

3.2.3.6. Antarmuka Detail Informasi Wisata pada Android

Pada Gambar 3.26 dapat dilihat rancangan antarmuka *detail* destinasi wisata dari *card list* atau *marker* sebelumnya yang telah di klik yang berisi informasi lengkap, berupa foto, pin lokasi, jarak dari lokasi, *rating* dan deskripsi meliputi jam buka, harga tiket masuk, dan informasi lainnya terkait wisata tersebut.



Gambar 3.25 Desain Antar Muka Marker Peta Destinasi



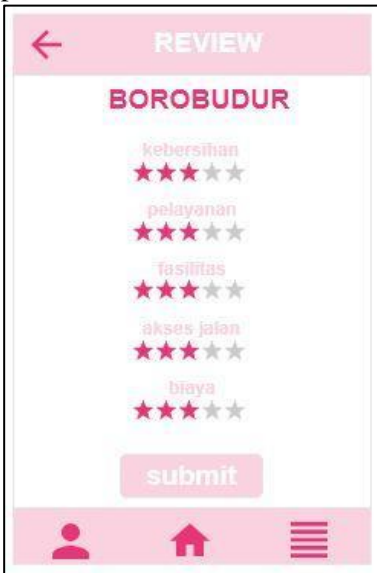
Gambar 3.26 Desain Antar Muka Detail Informasi Wisata

3.2.3.7. Antarmuka Fitur Review Destinasi Wisata pada Android

Pada Gambar 3.27 dapat dilihat rancangan antarmuka *review* destinasi wisata, dimana pengunjung dapat memberikan rating serta komentar terkait destinasi wisata yang telah dikunjungi.

3.2.3.8. Antarmuka Edit Profile Pengguna pada Android

Pada Gambar 3.28 dapat dilihat rancangan antarmuka *profile* pengguna yang berisi foto profil, nama, *username*, no hp, email dan password.



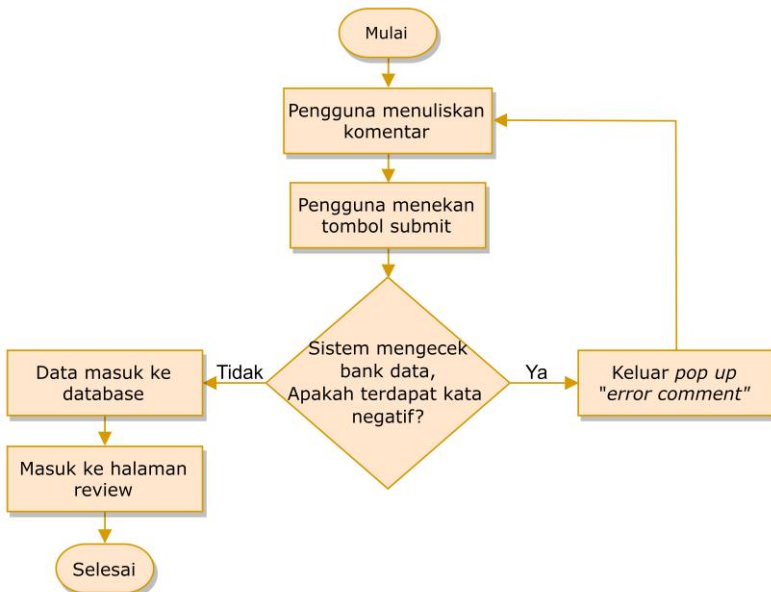
Gambar 3.27 Desain Antar Muka Fitur Review



Gambar 3.28 Desain Antar Muka Fitur Edit Profile

3.2.4. Perancangan Proses Kontrol Komentar Negatif

Proses ini bertujuan untuk melakukan pengawasan atas komentar negatif, pengawasan dilakukan oleh administrator dengan cara pembuatan bank data dimana bank data tersebut berisi kata-kata yang tidak boleh ditampilkan pada kolom *review*, sehingga ketika ada pengguna yang *submit* komentar dan mengandung kata-kata yang terdapat pada bank data maka sistem akan memberikan pemberitahuan kepada pengguna bawasannya ulasan yang diberikan mengandung kata-kata yang tidak seharusnya yang secara otomatis sistem tidak akan *submit* ulasan tersebut. Adapun proses dapat dilihat pada digram alir pada Gambar 3.29. Selain itu, administrator juga tetap melakukan pengecekan secara manual, dengan cara mengecek satu per satu hasil *review* dari pengguna dan ketika ditemukan *review* dengan kata-kata negatif maka administrator akan menghapus komentar tersebut melalui *website* administrator.



Gambar 3.29 Diagram Alir Proses Kontrol komentar negatif

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari analisis dan perancangan sistem yang telah dibahas pada Bab III. Namun dalam penerapannya, rancangan tersebut dapat mengalami perubahan minor sewaktu-waktu apabila dibutuhkan.

4.1. Lingkungan Implementasi

Dalam proses perancangan sistem ini digunakan perangkat pendukung sebagai berikut.

4.1.1. Lingkungan Implementasi Perangkat Keras

Tabel 4.1 menjelaskan lingkungan perangkat keras yang digunakan dalam proses pengembangan sistem perangkat lunak.

Tabel 4.1 Lingkungan Implementasi Perangkat Keras

<i>Hardware</i>	Laptop Dell, 2.00 Ghz <i>processor</i> Intel Core i7, 8 GB RAM
<i>Server</i>	512 MB RAM, Ubuntu 14.
<i>Smartphone</i>	Android dengan versi sistem operasi minimum 7.0

4.1.2. Lingkungan Implementasi Perangkat Lunak

Pada Tabel 4.2 menjelaskan implementasi perangkat lunak yang digunakan dalam proses pengembangan dan implementasi aplikasi ini.

Tabel 4.2 Lingkungan Implementasi Perangkat Lunak

Nama Aplikasi	Kegunaan
Microsoft Windows 10	Sebagai sistem operasi pada perangkat keras pengembangan aplikasi
Ubuntu 14	Sebagai sistem operasi pada <i>server</i>
MySQL	Sebagai basis data pada pengembangan aplikasi.
PHP 7.0	Sebagai bahasa pemrograman yang digunakan sebagai <i>back-end</i> di <i>server</i>
Apache HTTP Server	Sebagai <i>web server</i> aplikasi

Nama Aplikasi	Kegunaan
Power Designer 15	Sebagai aplikasi perancangan basis data
Android Studio	Sebagai IDE untuk implementasi aplikasi
Andrid SDK	Sebagai tool untuk mengembangkan aplikasi android
Sublime 2	Sebagai teks editor untuk penulisan kode sumber

4.2. Lingkungan Implementasi Antarmuka

Pada subbab ini dibahas bagaimana implementasi anatrmuka berdasarkan rancangan antar muka yang telah dibahas pada Bab 3.

4.2.1. Antarmuka Splash Screen

Implementasi tampilan splash screen dapat dilihat pada Gambar 4.1, pada gambar menunjukan ikon utama wisata Magelang yaitu Candi Borobudur, splash screen berjalan selama 5 detik ketika aplikasi pertama kali dijalankan.



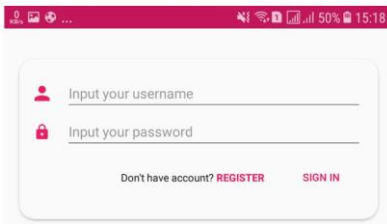
Gambar 4.1 Tampilan Splash Screen

4.2.2. Antarmuka Form Login pada Android

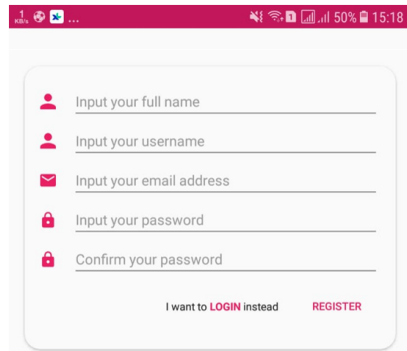
Implementasi form Login pada aplikasi Android ditunjukkan pada Gambar 4.2, pada form ini pengguna diminta untuk memasukkan *username* dan *password* yang sebelumnya telah diregistrasikan, setelah pengguna berhasil *sign in* maka pengguna akan masuk pada fitur *filter* kategori dan jika pengguna gagal maka sistem akan memunculkan pemberitahuan berupa *pop up* (*snack break*) bahwa *username* atau *password* tidak sesuai.

4.2.3. Antarmuka Form Register pada Android

Implementasi form register pada aplikasi Android ditunjukkan pada Gambar 4.3, pada form ini pengguna diminta untuk mengisi seluruh data, berupa *full name* (nama lengkap), *username* (nama pengguna), *email address* (alamat email), dan *password* (kata sandi). Email maupun *username* yang telah terdaftar tidak dapat didaftarkan kembali. Setelah berhasil registrasi sistem akan kembali pada tampilan *sign in*.



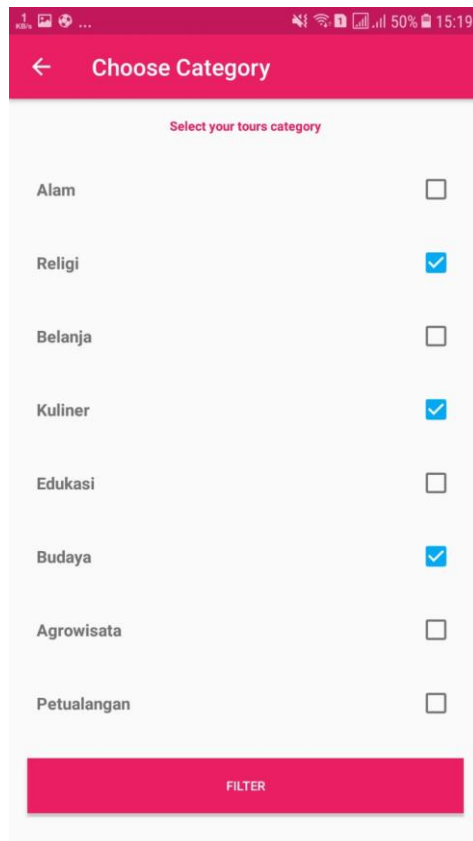
Gambar 4.2 Tampilan Form Login pada Android



Gambar 4.3 Tampilan Form Register pada Android

4.2.4. Antarmuka Fitur Filter Kategori pada Android

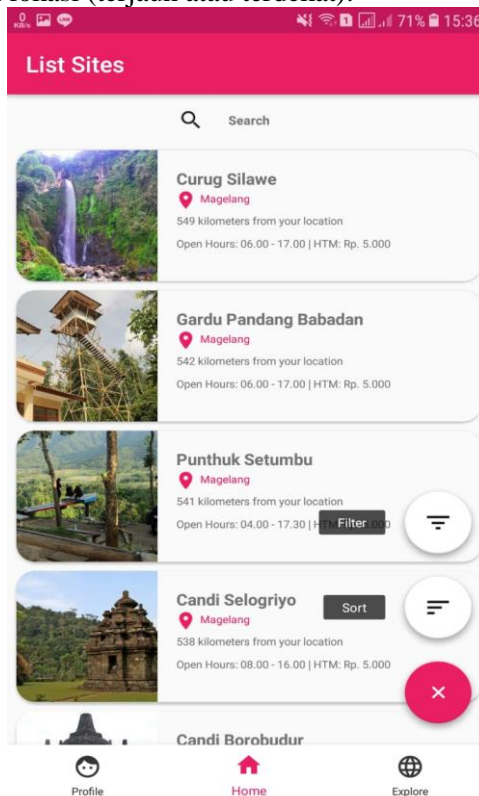
Implementasi Fitur Filtering pada aplikasi Android ditunjukkan pada Gambar 4.4. Sistem menampilkan *checkbox* daftar kategori wisata, dimana pengguna bisa memilih lebih dari satu kategori, setelah selesai memilih kategori yang diinginkan, sistem akan menampilkan hasil *filter* sesuai kategori yang telah dipilih sebelumnya.



Gambar 4.4 Tampilan Fitur Filter Kategori pada Android

4.2.5. Antarmuka List Hasil Filter Kategori pada Android

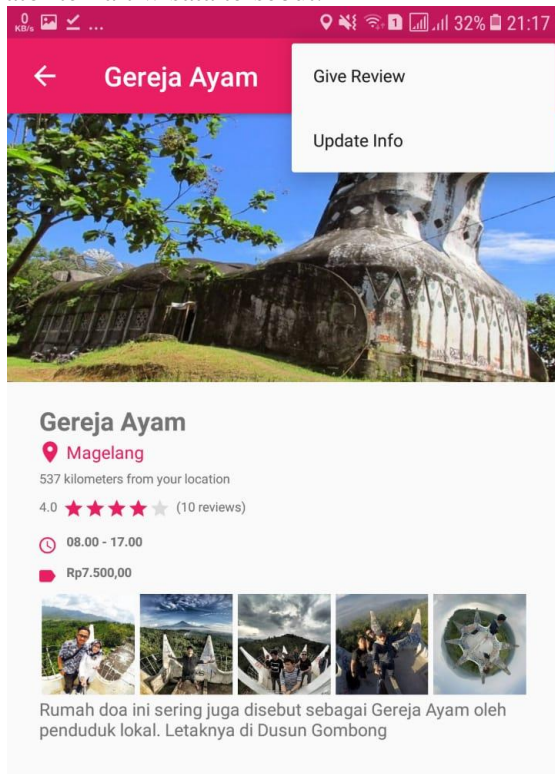
Implementasi Tampilan list hasil filter kategori pada aplikasi Android ditunjukkan pada Gambar 4.5. Sistem menampilkan daftar destinasi wisata berupa gambar, nama tempat wisata, jarak dari lokasi pengguna dan informasi singkat terkait destinasi wisata tersebut dan pada bagian bawah kanan terdapat tombol *Floating Action Button* yang terdiri dari *filter* untuk masuk ke halaman fitur *filter* kategori dan *sort* untuk mengurutkan konten wisata berdasarkan lokasi (terjauh atau terdekat).



Gambar 4.5 Tampilan Card List Destinasi Wisata pada Android

4.2.6. Antarmuka Detail Informasi Wisata pada Android

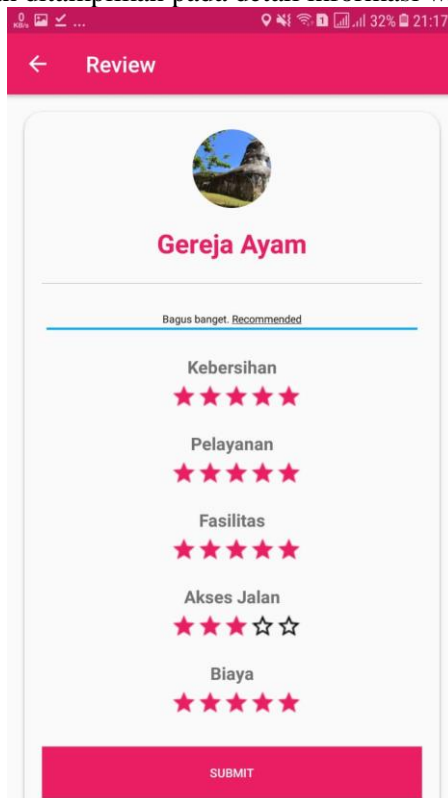
Implementasi Tampilan Detail informasi wisata pada aplikasi Android ditunjukkan pada Gambar 4.6 Detail informasi wisata meliputi rating dari hasil review para pengguna, deskripsi singkat terkait wisata termasuk didalamnya jam buka dan harga tiket masuk kemudian lokasi wisata berupa pin *map*, dimana ketika pin *map* di klik akan menampilkan peta lokasi wisata dan pada *option menu bar* sebelah kanan atas digunakan untuk pengguna yang ingin memberikan review ataupun *update* informasi kepada administrator terkait wisata tersebut.



Gambar 4.6 Tampilan Detail Informasi Wisata pada Android

4.2.7. Antarmuka Fitur Review Destinasi Wisata pada Android

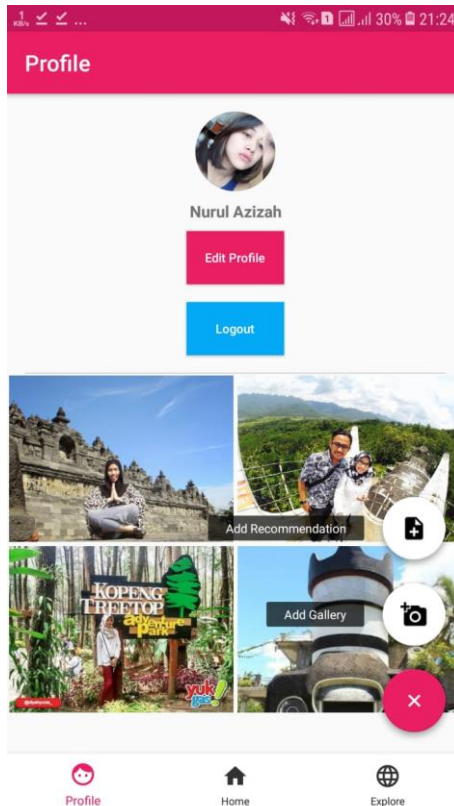
Implementasi Fitur *Review* destinasi wisata pada aplikasi Android ditunjukkan pada Gambar 4.7. Pada fitur *review* pengguna dapat memberikan *review* terkait destinasi wisata tersebut baik itu komentar maupun *rating bar* meliputi kebersihan, pelayanan, akses jalan, biaya dan fasilitas selain itu pengguna juga dapat melihat hasil *review* pengguna lain. Hasil dari *review* pada *rating bar* akan di *average* dan ditampilkan pada detail informasi wisata nantinya.



Gambar 4.7 Tampilan Fitur Review Destinasi Wisata pada Android

4.2.8. Antarmuka Melihat Profile Pengguna pada Android

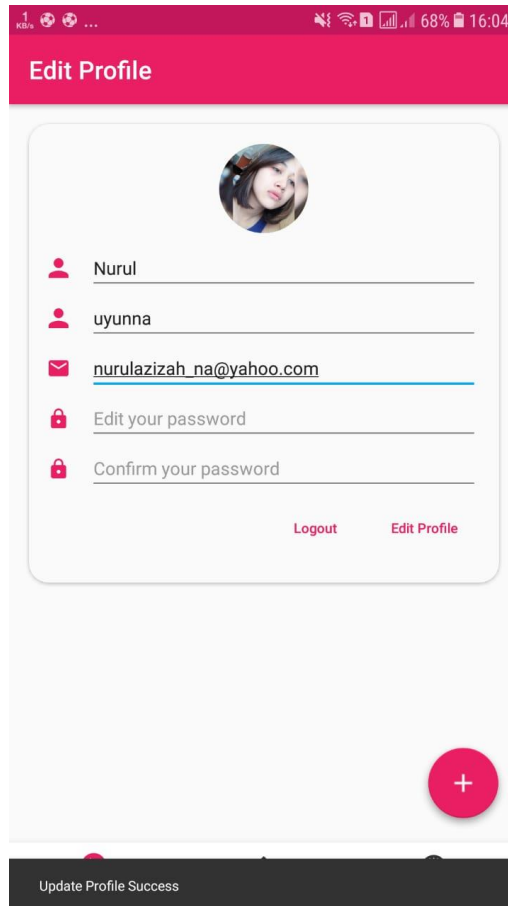
Implementasi Tampilan untuk melihat profil pengguna pada aplikasi Android ditunjukkan pada Gambar 4.8. *profile* pengguna berisi foto profil, galeri foto, tombol *edit profile*, *floating action button* untuk menambah rekomendasi wisata dan untuk unggah foto galeri spot wisata.



Gambar 4.8 Tampilan Melihat Profile Pengguna pada Android

4.2.9. Antarmuka Edit Profile Pengguna pada Android

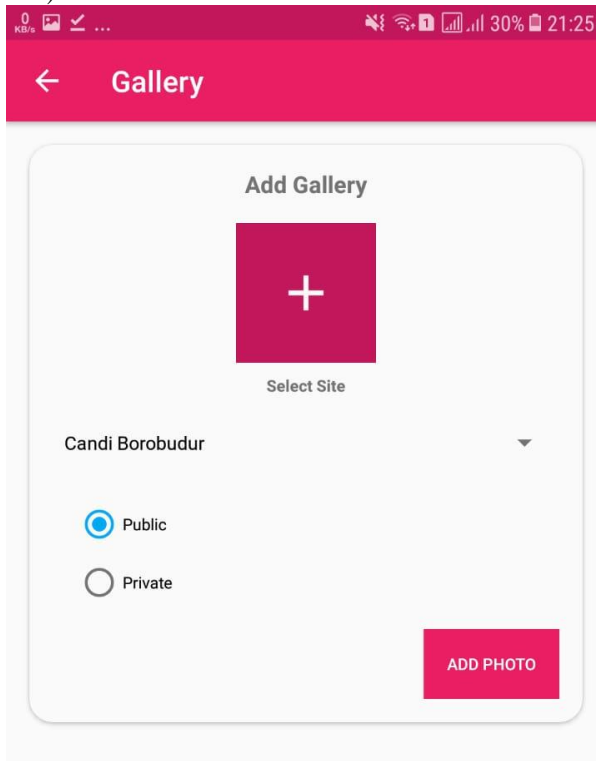
Implementasi Tampilan untuk mengedit profil pengguna pada aplikasi Android ditunjukkan pada Gambar 4.9. *profile* pengguna berisi foto profil yang dapat diunggah dari *device*, nama, *username*, email dan *password*, sistem akan menampilkan *popup* pemberitahuan apabila data telah berhasil terupdate.



Gambar 4.9 Tampilan Edit Profile Pengguna pada Android

4.2.10. Antarmuka Halaman Add Gallery pada Android

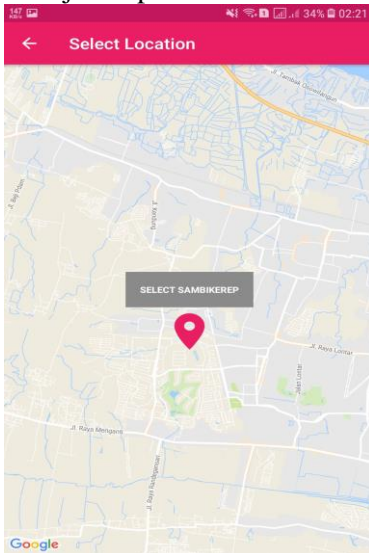
Implementasi Tampilan untuk melakukan unggah foto galeri spot wisata pada aplikasi Android ditunjukkan pada Gambar 4.10. halaman *Gallery* berisikan kotak unggah foto, dropdown pilihan tempat wisata dan *checkbox* pilihan pengunggahan foto (*private* atau *public*).



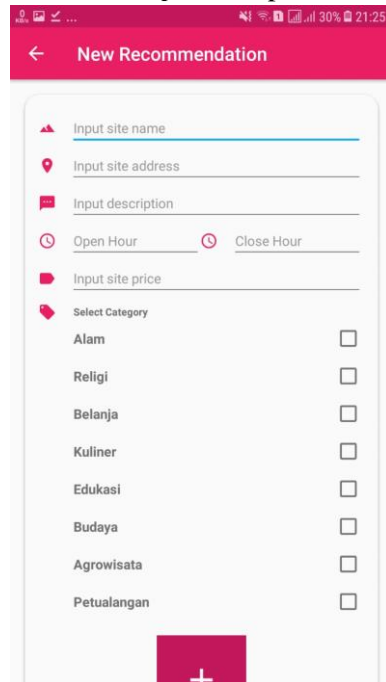
Gambar 4.10 Tampilan Halaman Add Gallery pada Android

4.2.11. Antarmuka Tambah Rekomendasi Wisata pada Android

Implementasi Fitur untuk pengguna menambahkan rekomendasi destinasi wisata pada aplikasi Android ditunjukkan pada Gambar 4.12 Pengguna harus mengisi seluruh data meliputi nama wisata, alamat, deskripsi, jam operasional, harga tiket masuk, kotak unggah foto, mengisi kategori berupa *checkbox* yang dapat dipilih lebih dari satu begitupun untuk menambah informasi lokasi wisata pengguna bisa mengklik tombol *click here to input location* yang nantinya akan dipindah ke halaman *map* untuk pin lokasi, ditunjukan pada Gambar 4.11



Gambar 4.11 Tampilan Input Location pada fitur Rekomendasi Wisata pada Android



Gambar 4.12 Tampilan Tambah Rekomendasi Wisata pada Android

4.2.12. Antarmuka Halaman Update Informasi Wisata pada Android

Implementasi Tampilan untuk melakukan *update* informasi wisata pada Android ditunjukkan pada Gambar 4.10. halaman *Update Information* berisikan form yang sama persis dengan halaman rekomendasi wisata baru, pengguna hanya perlu memilih pada bagian mana yang perlu diperbarui.

Update Information

Kopeng Treetop Adventure Park

null

Seperti namanya Kopeng Treetop Adventure merupakan salah satu objek wisata alam yang memanfaatkan ta

Open Hour Close Hour

Input site price

Select Category

Alam ☐

Religi ☐

Belanja ☐

Kuliner ☐

Edukasi ☐

Budaya ☐

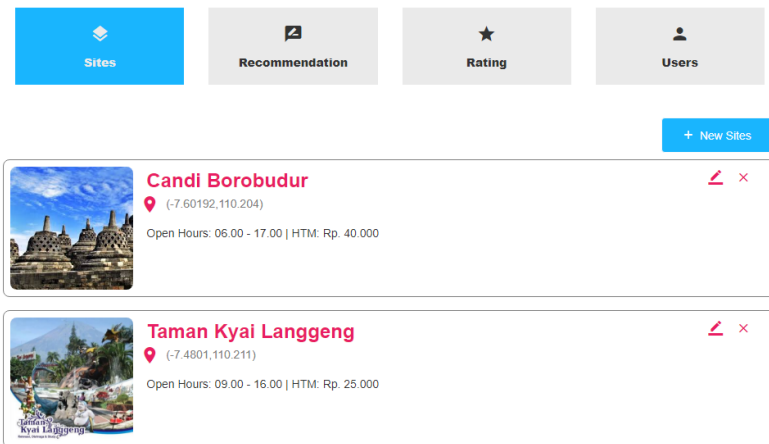
Agrowisata ☐

Petualangan ☐

Gambar 4.13 Tampilan Halaman Update Informasi Wisata pada Android

4.2.13. Antarmuka Kelola Wisata pada Web Administrator

Implementasi Fitur Kelola wisata pada Web Administrator ditunjukkan pada Gambar 4.14. Kelola wisata berisi daftar site wisata dimana didalam *card* terdapat informasi mengenai nama site, lokasi dan deskripsi singkat. Disini administrator dapat melakukan kelola wisata berupa tambah, *edit* dan hapus.



Gambar 4.14 Tampilan Kelola Wisata pada Web Administrator

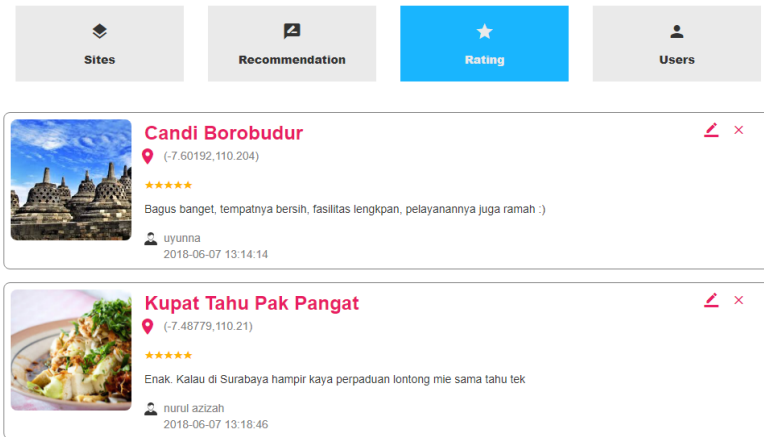
4.2.14. Antarmuka Kelola Rekomendasi Wisata pada Web Administrator

Kelola rekomendasi wisata berisi daftar site wisata dimana didalam *card* terdapat informasi mengenai nama site, lokasi dan deskripsi singkat. Disini administrator dapat melakukan kelola rekomendasi wisata berupa terima dan hapus.

4.2.15. Antarmuka Kelola Review Wisata pada Web Administrator

Implementasi Fitur Kelola review destinasi wisata pada Web Administrator ditunjukkan pada Gambar 4.15. Kelola *review* wisata berisi daftar site wisata dimana didalam *card* terdapat informasi mengenai nama site, lokasi, hasil *review* dan *rating* dari

pengguna serta nama dan foto pengguna yang memberikan *review* beserta waktu dan tanggal *submit review*. Disini administrator dapat melakukan kelola *review* wisata berupa tambah dan hapus.



Gambar 4.15 Tampilan Kelola Review Wisata pada Web Administrator

4.2.16. Antarmuka Kelola Profile Pengguna pada Web Administrator

Kelola rekomendasi wisata berisi daftar pengguna dimana didalam *card* terdapat informasi mengenai foto profil, nama, *username*, dan *email*. Disini administrator dapat melakukan manajemen pengguna wisata berupa tambah, *edit* dan hapus.

4.3. Implementasi Basis Data

Pada subbab ini akan dibahas secara mendetail rancangan basis data yang telah dipaparkan pada Bab 3. Pada implementasi ini dijelaskan mengenai pembuatan tabel-tabel yang digunakan pada sistem. Implementasi tabel basis data menggunakan sintaks MySQL.

4.3.1. Implementasi Tabel Site

Tabel *Site* adalah tabel yang digunakan untuk menyimpan data destinasi wisata. Implementasi tabel *Site* dapat dilihat pada Kode Sumber 4.1 . Tabel *Site* memiliki atribut yaitu *latitude* dan *longitude* sebagai koordinat lokasi destinasi wisata, *name* sebagai nama destinasi wisata, *description* sebagai deskripsi informasi mengenai destinasi wisata terkait dan *imageurl* untuk menyimpan url gambar . *site_id* sebagai *primary key*.

```
CREATE TABLE `tb_site` (
  `id` int(11) NOT NULL,
  `name` varchar(150) NOT NULL,
  `latitude` float NOT NULL,
  `longitude` float NOT NULL,
  `description` longtext NOT NULL,
  `image` text,
  `address` varchar(500) DEFAULT NULL,
  `open` varchar(10) NOT NULL DEFAULT '08.00',
  `close` varchar(10) NOT NULL DEFAULT '17.00',
  `price` double NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Kode Sumber 4.1 Implementasi Tabel Site

4.3.2. Implementasi Tabel Recommendation

Tabel *Recommendation* adalah tabel yang digunakan untuk menyimpan data rekomendasi dari pengguna. Tabel *Recommendation* memiliki beberapa atribut sebagai berikut: *rec_id* sebagai *primary key*, *latitude* dan *longitude*, *sites_name* dan *recommendation* serta *user_id* sebagai *foreign key*. Implementasi tabel *Recommendation* ditunjukkan pada Kode Sumber 4.2.

```
CREATE TABLE `tb_recommendation` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `name` varchar(100) NOT NULL,
  `latitude` float NOT NULL,
  `longitude` float NOT NULL,
  `recommendation` varchar(500) NOT NULL,
  `timestamp` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP,
```

```

`thumbnail` varchar(500) NOT NULL DEFAULT
'http://icons.iconarchive.com/icons/paomedia/small-n-
flat/256/house-icon.png',
`address` varchar(500) DEFAULT NULL,
`category_id` int(11) NOT NULL,
`multicategory` varchar(100) DEFAULT NULL,
`editmode` tinyint(1) DEFAULT '0',
`open` varchar(10) NOT NULL DEFAULT '08.00',
`close` varchar(100) NOT NULL DEFAULT '17.00',
`price` double NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Kode Sumber 4.2 Implementasi Tabel Recommendation

4.3.3. Implementasi Tabel Review

Tabel *Review* merupakan tabel yang digunakan untuk menyimpan hasil review dari pengguna. Atribut pada tabel ini adalah *rating*, *comment*, kemudian *rev_id* sebagai *primary key* dan *site_id* serta *user_id* sebagai *foreign key*. Implementasi ditunjukkan pada Kode Sumber 4.3.

```

CREATE TABLE `tb_review` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `site_id` int(11) NOT NULL,
  `comment` varchar(500) DEFAULT NULL,
  `timestamp` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP,
  `sanity` float DEFAULT '0',
  `facility` float DEFAULT '0',
  `access` float DEFAULT '0',
  `service` float DEFAULT '0',
  `cost` float DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Kode Sumber 4.3 Implementasi Tabel Review

4.3.4. Implementasi Tabel User

Tabel *user* merupakan tabel yang digunakan sebagai penyimpanan data *user*. Atribut pada tabel *users* adalah *user_id* sebagai *primary key* kemudian *email*, *password*, *username*, *fullname*, *imageurl*, *level*, *session*, dan *expired*. Implementasi tabel ini dapat dilihat pada Kode Sumber 4.4.

```

CREATE TABLE `tb_user` (
  `id` int(11) NOT NULL,

```

```

`username` varchar(20) NOT NULL,
`email` varchar(100) NOT NULL,
`password` varchar(45) NOT NULL,
`fullname` varchar(100) NOT NULL,
`image` varchar(500) DEFAULT
'https://cdn1.iconfinder.com/data/icons/rcons-user-
action/512/user-256.png',
`level` int(11) NOT NULL DEFAULT '0',
`session` varchar(45) DEFAULT NULL,
`expired` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Kode Sumber 4.4 Implementasi Tabel Users

4.3.5. Implementasi Tabel Category

Tabel *category* merupakan tabel yang digunakan sebagai penyimpanan data *category*. Atribut pada tabel *category* adalah *kat_id* sebagai *primary* dan nama *kategori*. Implementasi tabel ini dapat dilihat pada Kode Sumber 4.5.

```

CREATE TABLE `tb_category` (
  `id` int(11) NOT NULL,
  `category_name` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Kode Sumber 4.5 Implementasi Tabel Category

4.3.6. Implementasi Tabel Site_Category

Tabel *site_category* merupakan tabel yang digunakan sebagai penghubung antara tabel kategori dan site. Atribut pada tabel *site_category* adalah *kat_id* dan *site_id* keduanya sebagai *foreign key*. Implementasi tabel ini dapat dilihat pada Kode Sumber 4.6.

```

REATE TABLE `tb_site_category` (
  `id` int(11) NOT NULL,
  `site_id` int(11) NOT NULL,
  `category_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Kode Sumber 4.6 Implementasi Tabel Site_Category

4.3.7. Implementasi Tabel Gallery

Tabel *user* merupakan tabel yang digunakan sebagai penyimpanan data *user*. Atribut pada tabel *users* adalah *user_id* sebagai *primary key* kemudian *email*, *password*, *username*,

fullname, *imageurl*, *level*, *session*, dan *expired*. Implementasi tabel ini dapat dilihat pada Kode Sumber 4.7.

```
CREATE TABLE `tb_gallery` (
  `id` int(11) NOT NULL,
  `site_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `filepath` text NOT NULL,
  `timestamp` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP,
  `public` tinyint(1) NOT NULL DEFAULT '1'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Kode Sumber 4.7 Implementasi Tabel Gallery

4.4. Implementasi Web Service

Pada subbab ini akan dibahas bagaimana implementasi *web service* dilakukan. *Web service* yang digunakan pada sistem ini menggunakan metode *representational state transfer* (REST) dengan *framework* ReactJS. Format data yang digunakan dalam proses pertukaran data adalah *JavaScript Object Notation* (JSON). Pada subbab ini akan dibahas implementasi dari proses *query* pada *web service*. Berikut ini adalah fungsi-fungsi yang akan dibangun pada *web service*.

Implementasi *API Endpoint* untuk Aplikasi Wisata Magelang dijelaskan pada Tabel 4.3 dibawah ini

Tabel 4.3 Implementasi API Endpoint

No.	Endpoint URI	Method	Deskripsi
Situs Wisata			
1	/controller/site/get.php	GET	Mendapatkan daftar situs wisata
Parameter kembalian : <ul style="list-style-type: none"> • success : parameter sukses atau tidak nya permintaan ke server • message : parameter pesan yang diberikan oleh server • <i>Array of JSON</i>: <ul style="list-style-type: none"> - site_id - name - latitude - longitude - description 			

<ul style="list-style-type: none"> - image - address - open - close - price 			
2	/controller/site/create.php	POST	Menambahka n situs wisata
Parameter yang dikirim : <ul style="list-style-type: none"> • name • latitude • longitude • description • image • address • open • close • price 			
Parameter kembalian : <ul style="list-style-type: none"> • success : parameter sukses atau tidak nya permintaan ke server • message : parameter pesan yang diberikan oleh server 			
3	/controller/site/delete.php	DELETE	Menghapus situs wisata
Parameter kembalian : <ul style="list-style-type: none"> • success : parameter sukses atau tidak nya permintaan ke server • message : parameter pesan yang diberikan oleh server 			
4	/controller/site/edit.php	PUT	Merubah informasi situs wisata
Parameter yang dikirim : <ul style="list-style-type: none"> • site_id • name • latitude • longitude • description • image • address • open 			

<ul style="list-style-type: none"> • close • price 			
Parameter kembalian : <ul style="list-style-type: none"> • success : parameter sukses atau tidak nya permintaan ke server • message : parameter pesan yang diberikan oleh server 			
Rekomendasi Wisata			
5	/controller/recommendation/get.php	GET	Mendapatkan daftar rekomendasi situs wisata dari pengguna
Parameter kembalian : <ul style="list-style-type: none"> • success : parameter sukses atau tidak nya permintaan ke server • message : parameter pesan yang diberikan oleh server • <i>Array of JSON</i>: <ul style="list-style-type: none"> - recommendation_id - user full name - user profile picture url - site recommendation name - site recommendation latitude - site recommendation longitude - recommendation text - timestamp - site recommendation picture - multicategory - editmode - open - close - price 			
6	/controller/recommendation/delete.php	DELETE	Menghapus rekomendasi situs wisata dari pengguna
Parameter kembalian : <ul style="list-style-type: none"> • success : parameter sukses atau tidak nya permintaan ke server • message : parameter pesan yang diberikan oleh server 			

Review Destinasi Wisata			
7	/controller/review/get.php	GET	Mendapatkan daftar review dari pengguna
Parameter kembalian : <ul style="list-style-type: none"> • success : parameter sukses atau tidak nya permintaan ke server • message : parameter pesan yang diberikan oleh server • <i>Array of JSON</i>: <ul style="list-style-type: none"> - review_id - user full name - user profile picture url - site name - site picture - rating - comment - timestamp 			
8	/controller/review/delete.php	DELETE	Menghapus review dari pengguna
Parameter kembalian : <ul style="list-style-type: none"> • success : parameter sukses atau tidak nya permintaan ke server • message : parameter pesan yang diberikan oleh server 			
Pengguna			
9	/controller/user/getalluser.php	GET	Mendapatkan daftar detail profile pengguna
Parameter kembalian : <ul style="list-style-type: none"> • success : parameter sukses atau tidak nya permintaan ke server • message : parameter pesan yang diberikan oleh server • <i>Array of JSON</i>: <ul style="list-style-type: none"> - user id - user full name - username - user profile picture - user level - user session 			

10	/controller/user/register.php	POST	Mengirim detail profile pengguna
Parameter yang dikirim : <ul style="list-style-type: none"> • username • user full name • email • password • level 			
Parameter kembalian : <ul style="list-style-type: none"> • success : parameter sukses atau tidak nya permintaan ke server • message : parameter pesan yang diberikan oleh server 			
11	/controller/user/delete.php	DELETE	Menghapus profile pengguna
Parameter kembalian : <ul style="list-style-type: none"> • success : parameter sukses atau tidak nya permintaan ke server • message : parameter pesan yang diberikan oleh server 			
12	/controller/user/sessioncheck.php	POST	Mengecek status <i>session</i>
Parameter kembalian : <ul style="list-style-type: none"> • success : parameter sukses atau tidak nya permintaan ke server • message : parameter pesan yang diberikan oleh server 			

4.5. Implementasi Proses Aplikasi

Pada sub bagian ini menjelaskan tentang proses-proses yang terjadi pada aplikasi. Proses-proses dari pengambilan data hingga pengiriman data ke *server* dari *smartphone*.

4.5.1. Implementasi Dependencies

Retrofit adalah library untuk request ke REST API. Implementasi instalasi pada *Retrofit* dapat dilihat Pada Kode Sumber 4.8.

1. implementation 'com.squareup.retrofit2:retrofit:2.4.0'
2. provided 'org.glassfish:javax.annotation:10.0-b28'
3. implementation 'com.google.code.gson:gson:2.7'

4. implementation 'com.squareup.retrofit2:converter-gson:2.1.0'
5. implementation 'com.squareup.okhttp3:logging-interceptor:3.9.1'

Kode Sumber 4.8 Implementasi Dependencies Retrofit

Glide adalah *library* untuk manajemen pemrosesan gambar ke dalam elemen *imageView*. *Library* ini juga dapat digunakan untuk mengambil gambar dari URL dan terdapat juga *cache* didalamnya sehingga mengurangi beban ketika *load image*. Implementasi instalasi *Glide* dapat dilihat pada Kode Sumber 4.9.

6. Implementation 'com.github.bumptech.glide:glide:4.1.1'
7. annotationProcessor 'com.github.bumptech.glide:compiler:4.6.1'
8. implementation 'com.github.bumptech.glide:glide:4.1.1'

Kode Sumber 4.9 Implementasi Dependencies Glide

Google Maps Services adalah *service google* yang digunakan untuk mendapatkan lokasi saat ini, menampilkan map, dan mendapatkan nama tempat dari lokasi saat ini. Implementasi *Google Maps Services* dapat dilihat pada Kode Sumber 4.10

9. implementation 'com.google.android.gms:play-services-location:15.0.0'
10. implementation 'com.google.android.gms:play-services-maps:15.0.0'
11. implementation 'com.google.android.gms:play-services-places:15.0.0'

Kode Sumber 4.10 Implementasi Dependencies Google Maps Service

ButterKnife adalah *library* untuk mempermudah binding elemen xml ke Java. Implementasi *ButterKnife* dapat dilihat pada Kode Sumber 4.11 .

12. implementation 'com.jakewharton:butterknife:8.8.1'
13. annotationProcessor 'com.jakewharton:butterknife-compiler:8.8.1'

Kode Sumber 4.11 Implementasi Dependencies ButterKnife

4.5.2. Implementasi Pengiriman Data

Pengiriman data *event* dari *devices* ke *server* menggunakan *library* Retrofit. Retrofit adalah HTTP *library* yang dapat melakukan perbaikan, pembaharuan, pemindahan sistem kontrol, mesin sehingga suatu sistem tersebut beroperasi lebih baik lagi. digunakan untuk pengiriman jaringan data.

Pada proses pengiriman data, terdapat suatu data yang dikirimkan ke *server*. Dalam implementasinya di kode sumber, data yang dikirim diberi nama *Payload*. Selain *payload*, khusus untuk pengiriman data dengan metode *GET*, terdapat data yang dikirim berupa *Query*. Setelah data *payload* atau *query* ini dikirim, maka *server* akan mengirimkan data balikan berupa informasi-informasi tertentu. Dalam implementasinya di kode sumber, data yang diterima ini diberi nama *Response*.

Pada sumber kode, masing-masing fungsi untuk setiap permintaan ke *server* dibedakan tergantung *payload* / *query*, metode permintaan, dan *response* dari *server*. Semua fungsi-fungsi ini diletakkan pada satu file bernama *RestInterface.java*. Pada Kode Sumber 4.12 terdapat contoh implementasi dari permintaan data dengan masing-masing tergantung *payload* / *query*, metode permintaan, dan *response* dari *server* yang berbeda-beda.

Data yang dikirimkan dalam bentuk format *JSON*. Kode Sumber 4.12 adalah implementasi fungsi untuk memasukkan data-data yang akan dikirimkan berupa *JSONObject*.

```
Interface RestInterface
1.  @POST("URL")
2.  @Headers("Content-Type")
3.  Call <ResponseName> nameResponse (@Body payloadName
    namePayload);
4.
5.  @GET("site/get.php")
    Call<GetAllSitesResponse.Sites>
6.  @Headers("Content-Type: application/json")
7.  getSite(@Query("id") int id);
```

Kode Sumber 4.12 Implementasi Pengiriman Data

4.5.3. Implementasi Pengambilan Data

Pengambilan data dilakukan dengan melakukan permintaan ke *server* melalui *Retrofit*. Pada *Retrofit*, pengambilan data dilakukan pada fungsi *onResponse*, dimana fungsi *onResponse* mengharuskan dua parameter pada fungsi yaitu obyek *Response* dan obyek permintaan itu sendiri. Data balikan dari *server* dapat diambil dengan memanggil fungsi *body* dari obyek *Response*. Implementasi ini dapat dilihat pada Kode Sumber 4.13

```
onResponse ()
1.  @Override
2.  public void onResponse(Call<UploadPictureResponse>
   call, Response<UploadPictureResponse> response) {
3.  if(response.body().isSuccess()) {
4.  PreferenceUtilities.setImage(getContext(),
   response.body().getFilepath());
5.  updateProfile();
6.  }
```

Kode Sumber 4.13 Implementasi Mengambil Data

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini dibahas tentang uji coba dan evaluasi pada perangkat lunak yang dibangun pada tugas akhir ini. Pengujian dilakukan pada fungsionalitas sistem perangkat lunak. Pada bab ini juga membahas mengenai dasar pengujian, pengujian fungsionalitas, pengujian akurasi, analisa data dan evaluasi pengujian.

5.1. Lingkungan Uji Coba

Dalam proses pengujian perangkat lunak, dibutuhkan suatu lingkungan uji coba yang sesuai dengan standar kebutuhan. Uji coba aplikasi ini dilakukan dengan menggunakan *smartphone* Android. Berikut ini spesifikasi *smartphone* yang digunakan.

- Merek *smartphone* Samsung J7 Prime
- Sistem operasi Android Nougat 7.0
- *Processor* Octa-core 1.6 GHz Cortex-A53
- Internal Storage 32 GB
- RAM 3 GB

5.2. Uji Coba Fungsionalitas

Pada subbab ini akan dijelaskan mengenai skenario uji coba yang dilakukan. Terdapat beberapa skenario uji coba yang telah dilakukan yaitu:

Aplikasi berbasis Android untuk Pengguna

1. Skenario pengujian 1 : Melakukan registrasi
2. Skenario pengujian 2 : Melakukan filter kategori pada menu Home
3. Skenario pengujian 3 : Melihat list hasil filter kategori
4. Skenario pengujian 4 : Melihat destinasi wisata secara keseluruhan dalam bentuk list
5. Skenario pengujian 5 : Melihat destinasi wisata secara keseluruhan dalam bentuk marker pada peta
6. Skenario pengujian 6 : Melihat detail informasi wisata

7. Skenario pengujian 7 : Memberikan review terkait wisata yang dipilih
8. Skenario pengujian 8 : Memberikan informasi terbaru terkait destinasi wisata yang sudah ada kepada administrator
9. Skenario pengujian 9 : Memberikan rekomendasi wisata baru kepada administrator
10. Skenario pengujian 10 : Melihat profile pengguna
11. Skenario pengujian 11 : Mengedit profile pengguna
12. Skenario pengujian 12 : Melakukan unggah foto untuk galeri spot wisata

Aplikasi berbasis Website untuk Administrator

1. Skenario pengujian 13 : Mengelola konten destinasi wisata
2. Skenario pengujian 14 : Menanggapi rekomendasi pengguna
3. Skenario pengujian 15 : Melakukan manajemen pengguna
4. Skenario pengujian 16 : Kontroler komentar negatif

Uji coba Fungsionalitas akan dijelaskan pada setiap Tabel skenario pengujian dibawah ini:

5.2.1. Pengujian Fitur Registrasi

Tabel 5.1 Pengujian Fitur Registrasi

Kode Pengujian	SP-01
Nama Use Case	Melakukan Registrasi
Nomor	UC-01
Aktor	Pengguna
Kondisi Awal	Data baru belum masuk ke database
Alur Normal	Pengguna menekan tombol <i>create new account</i>
	Sistem menampilkan <i>form</i> berupa <i>full name, username, password</i> dan <i>email</i>
	Pengguna mengisi form input data
	Pengguna menekan tombol <i>create my account</i>
	Sistem memasukan data ke database
	Sistem menampilkan pesan bahwa data berhasil disimpan

Hasil yang Diharapkan	Data baru tersimpan di database
Hasil Pengujian	Berhasil

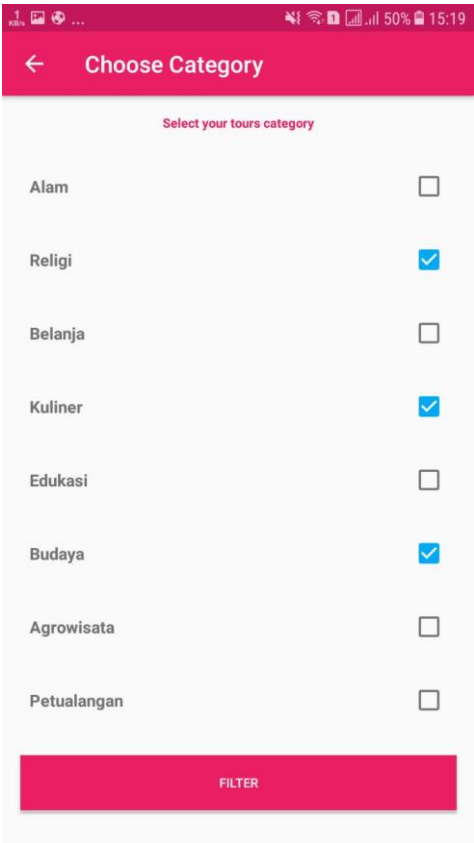
Gambar 5.1 Registrasi pada Android

5.2.2. Pengujian Filter Kategori pada menu Home

Tabel 5.2 Pengujian Filtering pada fitur pencarian

Kode Pengujian	SP-02
Nama Use Case	Melakukan filter kategori pada menu Home
Nomor	UC-02
Aktor	Pengguna
Kondisi Awal	Sistem menampilkan kriteria pencarian
Alur Normal	Pengguna menekan tombol home pada bagian tengah <i>bottom navigation view</i>
	Sistem menampilkan <i>checkbox</i> kategori pencarian
	Pengguna memilih kriteria yang diinginkan
	Pengguna menekan tombol <i>filter</i>
	Sistem menampilkan hasil <i>filter</i>

Hasil yang Diharapkan	Sistem memfilter sesuai kriteria yang dipilih
Hasil Pengujian	Berhasil



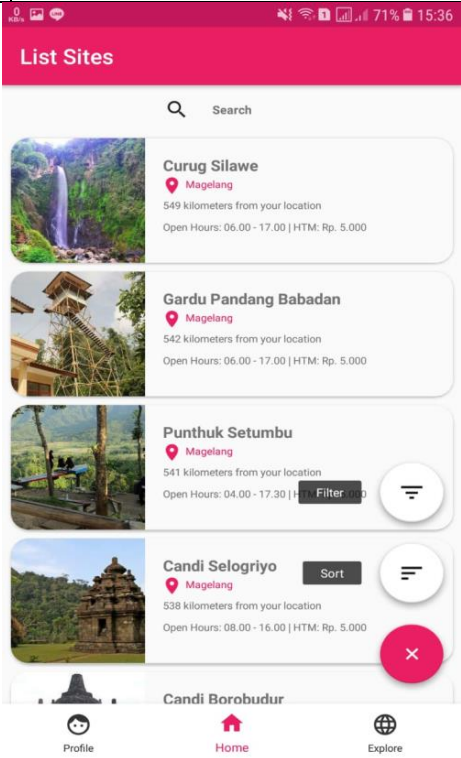
Gambar 5.2 Filter Kategori pada Android

5.2.3. Pengujian Melihat List hasil Filter Kategori

Tabel 5.3 Pengujian Melihat list hasil filtering dalam bentuk card

Kode Pengujian	SP-03
----------------	-------

Nama Use Case	Melihat list hasil filter kategori
Nomor	UC-03
Aktor	Pengguna
Kondisi Awal	Sistem tidak menampilkan hasil filtering
Alur Normal	Pengguna melakukan <i>filter</i> kategori terlebih dahulu sesuai alur UC-02
	Sistem menampilkan <i>list</i> hasil <i>filter</i>
Hasil yang Diharapkan	Sistem menampilkan list hasil filtering
Hasil Pengujian	Berhasil



Gambar 5.3 List hasil filter kategori pada Android

5.2.4. Pengujian Melihat destinasi wisata secara keseluruhan dalam bentuk list

Tabel 5.4 Pengujian Melihat destinasi wisata secara keseluruhan dalam bentuk list

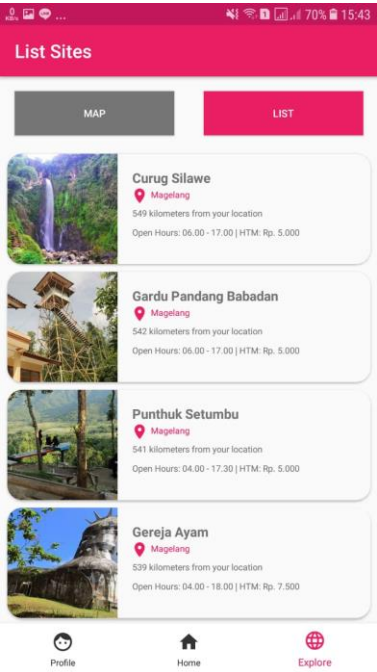
Kode Pengujian	SP-04
Nama Use Case	Melihat destinasi wisata secara keseluruhan dalam bentuk list
Nomor	UC-04
Aktor	Pengguna
Kondisi Awal	Sistem tidak menampilkan destinasi wisata
Alur Normal	Pengguna menekan tombol <i>Explore</i> pada <i>bottom navigation view</i>
	Sistem menampilkan seluruh destinasi wisata dalam bentuk <i>marker</i> pada peta
	Pengguna menekan tombol <i>List</i>
	Sistem menampilkan <i>list</i> seluruh destinasi wisata
Hasil yang Diaharapkan	Sistem menampilkan seluruh destinasi wisata dalam bentuk list
Hasil Pengujian	Berhasil

5.2.5. Pengujian Melihat destinasi wisata secara keseluruhan dalam bentuk marker pada peta

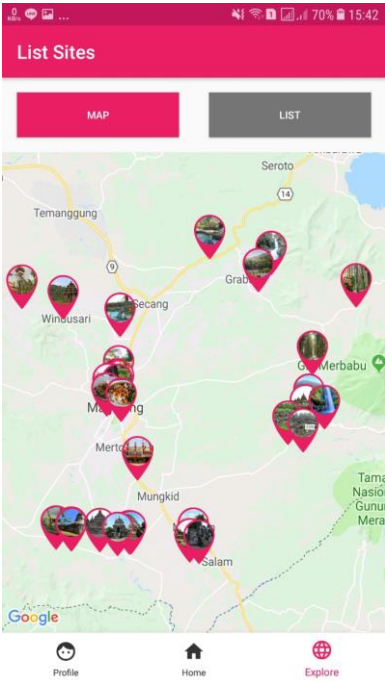
Tabel 5.5 Pengujian Melihat destinasi wisata secara keseluruhan dalam bentuk peta

Kode Pengujian	SP-05
Nama Use Case	Melihat destinasi wisata secara keseluruhan dalam bentuk marker peta
Nomor	UC-05
Aktor	Pengguna
Kondisi Awal	Sistem tidak menampilkan destinasi wisata
Alur Normal	Pengguna menekan tombol <i>Explore</i> pada <i>bottom navigation view</i>

	Sistem menampilkan seluruh destinasi wisata dalam bentuk <i>marker</i> pada peta
Hasil yang Diharapkan	Sistem menampilkan seluruh destinasi wisata dalam bentuk <i>marker</i> pada peta
Hasil Pengujian	Berhasil



Gambar 5.4 List wisata secara keseluruhan pada Android



Gambar 5.5 Map wisata secara keseluruhan pada Android

5.2.6. Pengujian Melihat Detail Informasi wisata

Tabel 5.6 Pengujian Melihat detail informasi wisata

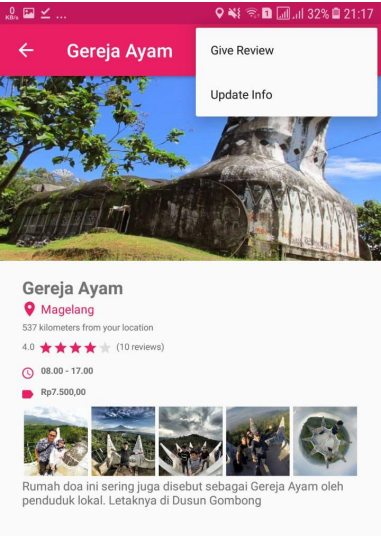
Kode Pengujian	SP-06
Nama Use Case	Melihat detail informasi wisata
Nomor	UC-06
Aktor	Pengguna
Kondisi Awal	Sistem menampilkan informasi singkat dalam bentuk <i>card</i> pada <i>list</i> atau <i>marker</i> pada peta
Alur Normal	Pengguna melakukan alur proses sesuai kasus penggunaan UC-03 / UC-04 / UC-05
	Pengguna menekan informasi singkat dalam bentuk <i>card</i> pada <i>list</i> / <i>marker</i> pada peta
	Sistem menampilkan detail informasi wisata
Hasil yang Diharapkan	Sistem menampilkan detail informasi
Hasil Pengujian	Berhasil

5.2.7. Pengujian Memberikan review terkait destinasi yang dipilih

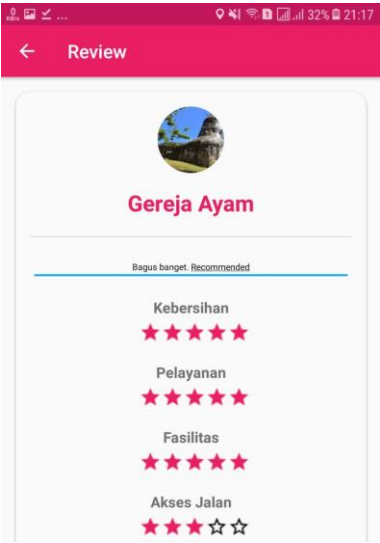
Tabel 5.7 Pengujian Memberikan review terkait destinasi yang dipilih

Kode Pengujian	SP-07
Nama Use Case	Memberikan review terkait destinasi wisata yang dipilih
Nomor	UC-07
Aktor	Pengguna
Kondisi Awal	Review wisata belum masuk ke database
Alur Normal	Pengguna melakukan alur proses sesuai kasus penggunaan UC-06
	Pengguna menekan tombol <i>option menu</i> pada bagian pojok atas kanan
	Sistem menampilkan pilihan <i>option menu</i>

	Pengguna memilih <i>give rating</i>
	Sistem menampilkan halaman <i>Review</i>
	Pengguna menekan tombol <i>Submit Review</i>
	Sistem menampilkan halaman <i>Rating and Review</i>
	Pengguna mengisi kolom komentar dan <i>rating bar</i>
	Pengguna menekan tombol <i>Submit</i>
	Sistem memasukan data ke database
	Sistem menampilkan pesan bahwa data berhasil disimpan
Hasil yang Diharapkan	Review tersimpan didatabase
Hasil Pengujian	Berhasil



Gambar 5.6 Detail Informasi wisata pada Android

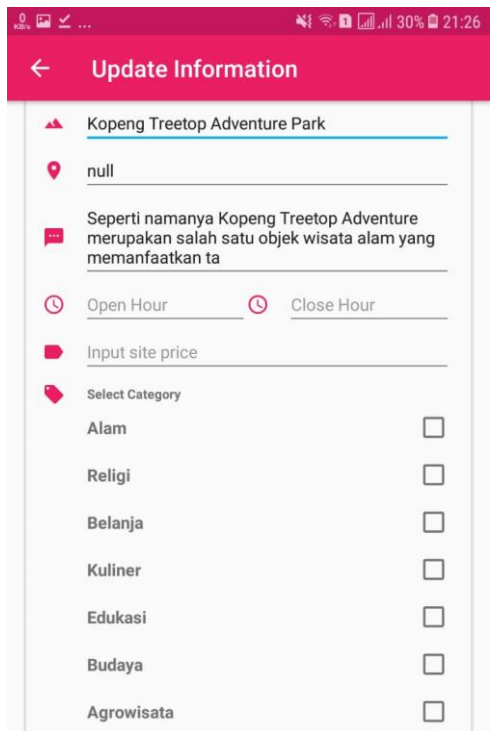


Gambar 5.7 Review wisata pada Android

5.2.8. Pengujian Fitur Memberikan informasi terbaru terkait destinasi wisata yang sudah ada kepada administrator

Tabel 5.8 Pengujian Fitur Memberikan informasi terbaru terkait destinasi wisata yang sudah ada kepada administrator

Kode Pengujian	SP-08
Nama Use Case	Memberikan informasi terbaru terkait destinasi wisata yang sudah ada kepada administrator
Nomor	UC-08
Aktor	Pengguna
Kondisi Awal	Sistem tidak menampilkan <i>profile</i>
Alur Normal	Pengguna melakukan alur proses sesuai kasus penggunaan UC-06
	Pengguna menekan tombol <i>option menu</i> pada bagian pojok atas kanan
	Sistem menampilkan pilihan <i>option menu</i>
	Pengguna memilih <i>recommendation update</i>
	Sistem menampilkan halaman <i>recommend an edit</i>
	Pengguna mengisi formulir
	Pengguna menekan tombol <i>Submit</i>
	Sistem memasukan data ke database
	Sistem menampilkan pesan bahwa data berhasil disimpan
Hasil yang Diharapkan	Data informasi wisata terbaru masuk ke database
Hasil Pengujian	Berhasil



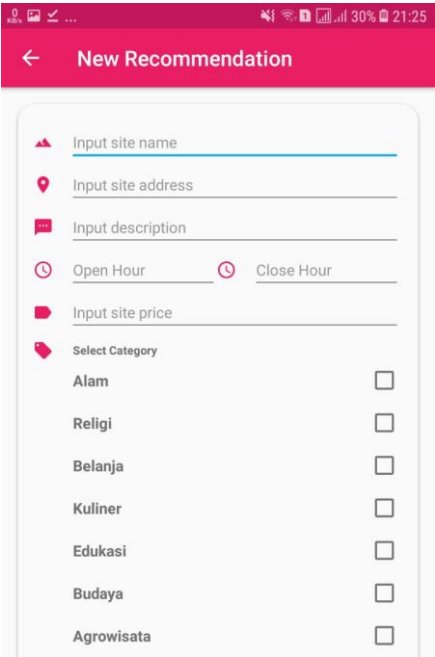
Gambar 5.8 Update Informasi Wisata pada Android

5.2.9. Pengujian Fitur Memberikan rekomendasi wisata baru kepada administrator

Tabel 5.9 Pengujian fitur memberikan rekomendasi wisata baru kepada administrator

Kode Pengujian	SP-09
Nama Use Case	Memberikan rekomendasi wisata baru kepada administrator
Nomor	UC-09
Aktor	Pengguna
Kondisi Awal	Data rekomendasi belum masuk ke database

Alur Normal	Pengguna menekan tombol menu <i>profile</i> pada <i>bottom navigation view</i>
	Sistem menampilkan <i>profile</i> pengguna
	Pengguna menekan tombol <i>Floating Action Button</i>
	Pengguna menekan memilih rekomendasi
	Sistem menampilkan formulir rekomendasi
	Pengguna mengisi formulir
	Pengguna menekan tombol <i>submit</i>
	Sistem memasukan data ke database
	Sistem menampilkan pesan bahwa data berhasil <i>diupload</i>
Hasil yang Diharapkan	Data rekomendasi masuk ke database
Hasil Pengujian	Berhasil



Gambar 5.9 Rekomendasi wisata baru pada Android

5.2.10. Pengujian Melihat Profile

Tabel 5.10 Pengujian Melihat Profile

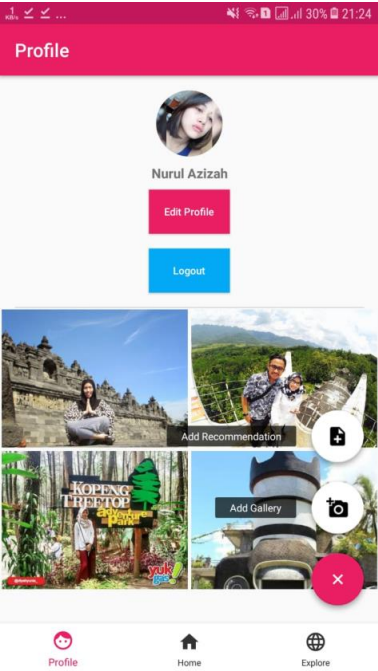
Kode Pengujian	SP-10
Nama <i>Use Case</i>	Melihat profile
Nomor	UC-10
Aktor	Pengguna
Kondisi Awal	Sistem tidak menampilkan <i>profile</i>
Alur Normal	Pengguna menekan tombol menu <i>profile</i> pada bagian kiri <i>buttom navigation view</i>
	Sistem menampilkan <i>profile</i> pengguna
Hasil yang Diharapkan	Sistem menampilkan <i>profile</i>
Hasil Pengujian	Berhasil

5.2.11. Pengujian Mengedit Profile

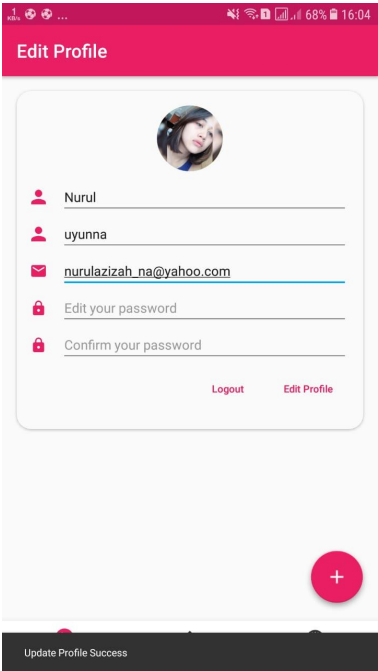
Tabel 5.11 Pengujian Mengedit Profile

Kode Pengujian	SP-11
Nama <i>Use Case</i>	Mengedit profile
Nomor	UC-11
Aktor	Pengguna
Kondisi Awal	Data belum <i>terupdate</i>
Alur Normal	Pengguna menekan tombol menu <i>profile</i> pada bagian kiri <i>buttom navigation view</i>
	Sistem menampilkan <i>profile</i> pengguna
	Pengguna mengubah isi pada form profile
	Pengguna menekan tombol <i>save</i>
	Sistem memperbarui data pada database
	Sistem menampilkan pesan bahwa data berhasil disimpan

Hasil yang Diharapkan	Update data baru tersimpan di database
Hasil Pengujian	Berhasil



Gambar 5.10 Melihat profile pengguna pada Android



Gambar 5.11 Edit Profile pengguna pada Android

5.2.12. Pengujian Melakukan unggah foto untuk galeri spot wisata

Tabel 5.12 Pengujian Melakukan unggah foto untuk galeri spot wisata

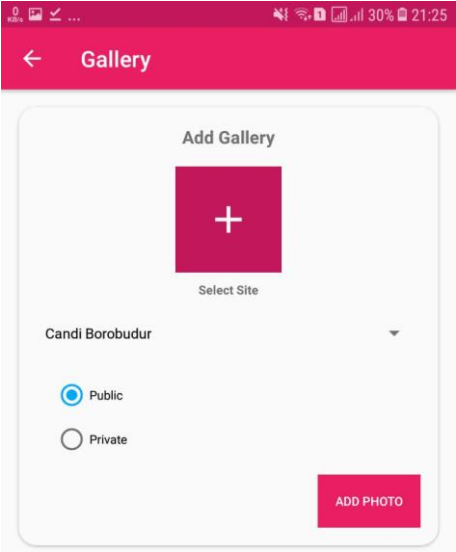
Kode Pengujian	SP-12
Nama Use Case	Melakukan unggah foto untuk galeri spot wisata
Nomor	UC-12
Aktor	Pengguna
Kondisi Awal	Data belum masuk ke database
Alur Normal	Pengguna menekan tombol menu <i>profile</i> pada <i>bottom navigation view</i>
	Sistem menampilkan <i>profile</i> pengguna
	Pengguna menekan tombol <i>Floating Action Button</i>
	Pengguna memilih <i>Add Gallery</i>
	Sistem menampilkan halaman <i>Add Gallery</i>
	Pengguna mengisi formulir
	Pengguna menekan tombol <i>Add photo</i>
	Sistem memasukan data ke database
	Sistem menampilkan pesan bahwa data berhasil diupload
Hasil yang Diharapkan	Data masuk ke database
Hasil Pengujian	Berhasil

5.2.13. Pengujian Mengelola konten wahana wisata

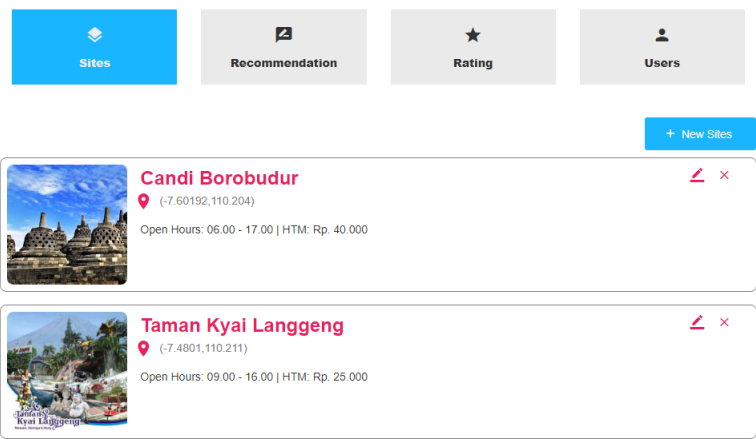
Tabel 5.13 Pengujian Mengelola wahan wisata

Kode Pengujian	SP-13
Nama Use Case	Mengelola konten wahana wisata
Nomor	UC-13
Aktor	Administrator

Kondisi Awal	Sistem belum menampilkan halaman kelola konten
Alur Normal	Administrator menekan tombol <i>menu</i> kelola konten
	Sistem menampilkan halaman kelola konten
	Administrator menekan tombol tambah data untuk menambahkan data baru atau menekan tombol <i>edit</i> / <i>delete</i> pada data yang dipilih untuk di <i>edit</i> / dihapus
	Sistem menampilkan form tambah data ketika administrator menekan tombol tambah data
	Administrator mengisi form
	Administrator menekan tombol <i>save</i>
	Sistem memperbarui data pada database
	Sistem menampilkan pesan bahwa data berhasil disimpan
Hasil yang Diharapkan	Sistem berhasil melakukan perubahan data sesuai perintah yang dijalankan
Hasil Pengujian	Berhasil



Gambar 5.12 Add Gallery pada Android



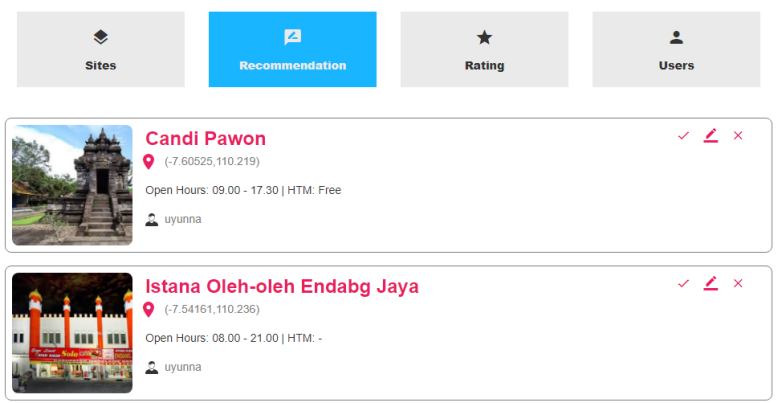
Gambar 5.13 Kelola Wisata pada Web Administrator

5.2.14. Pengujian Menanggapi rekomendasi pengguna

Tabel 5.14 Pengujian Menanggapi rekomendasi pengguna

Kode Pengujian	SP-14
Nama Use Case	Menanggapi rekomendasi pengguna
Nomor	UC-14
Aktor	Administrator
Kondisi Awal	Sistem belum menampilkan halaman rekomendasi wisata
Alur Normal	Administrator menekan tombol <i>menu</i> rekomendasi wisata
	Sistem menampilkan halaman rekomendasi wisata
	Administrator menekan tombol lihat untuk melihat ataupun mengedit data
	Sistem menampilkan form destinasi wisata
	Administrator mengedit isi form
	Administrator menekan tombol <i>upload</i>
	Sistem memperbarui data pada database
	Sistem menampilkan pesan bahwa data berhasil ditambahkan

Hasil yang Diharapkan	Sistem berhasil menambahkan data baru sesuai rekomendasi pengguna
Hasil Pengujian	Berhasil



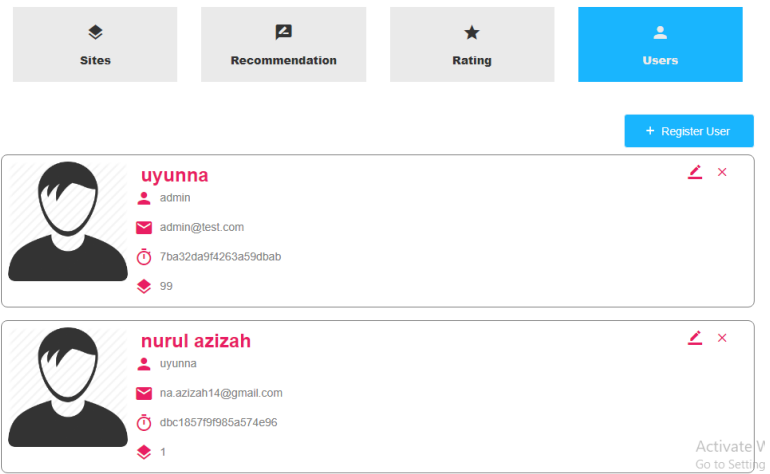
Gambar 5.14 Menanggapi rekomendasi wisata pada Web Administrator

5.2.15. Pengujian Melakukan manajemen pengguna

Tabel 5.15 Pengujian Melakukan manajemen pengguna

Kode Pengujian	SP-15
Nama Use Case	Melakukan manajemen pengguna
Nomor	UC-15
Aktor	Administrator
Kondisi Awal	Sistem belum menampilkan halaman manajemen pengguna
Alur Normal	Administrator menekan tombol <i>menu</i> manajemen pengguna
	Sistem menampilkan halaman manajemen pengguna
	Administrator menekan tombol lihat / <i>delete</i> pada data yang dipilih untuk di lihat / dihapus
	Sistem menampilkan data pengguna ketika administrator menekan tombol lihat dan sistem

	menampilkan modal konfirmasi ketika administrator menekan tombol <i>delete</i>
	Administrator menekan tombol <i>yes</i> pada modal untuk menghapus pengguna
	Sistem memperbarui data pada database
	Sistem menampilkan pesan bahwa data berhasil disimpan
Hasil yang Diharapkan	Sistem berhasil melakukan perubahan data sesuai perintah yang dijalankan
Hasil Pengujian	Berhasil



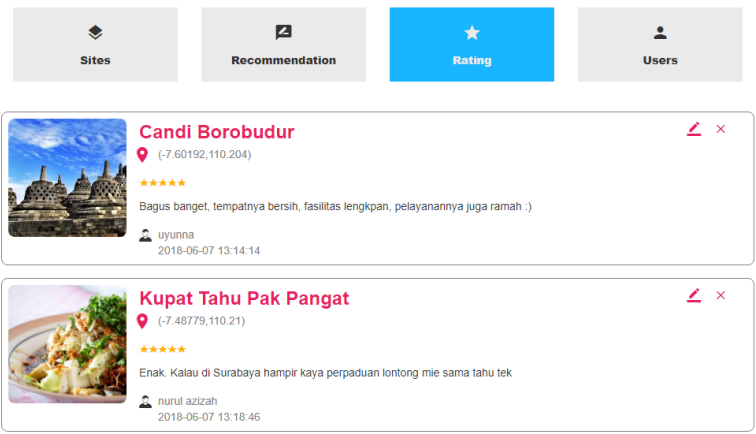
Gambar 5.15 Manajemen pengguna pada Web Administrator

5.2.16. Pengujian Kontrol komentar negatif

Tabel 5.16 Pengujian Kontrol komentar negatif

Kode Pengujian	SP-16
Nama Use Case	Kontroler komentar negatif
Nomor	UC-16
Aktor	Administrator

Kondisi Awal	Sistem belum menampilkan halaman moderator komentar
Alur Normal	Administrator menekan tombol <i>menu</i> moderator komentar
	Sistem menampilkan halaman moderator komentar
	Administrator menekan tombol <i>delete</i> pada data yang dipilih
	Sistem menampilkan modal konfirmasi ketika administrator menekan tombol <i>delete</i>
	Administrator menekan tombol <i>yes</i> pada modal untuk menghapus komentar
	Sistem memperbarui data pada database
	Sistem menampilkan pesan bahwa data berhasil dihapus
Hasil yang Diharapkan	Sistem berhasil melakukan penghapusan komentar negatif
Hasil Pengujian	Berhasil



Gambar 5.16 Kontrol komentar negatif pada Web Administrator

5.3. Uji Coba Response Time

Evaluasi Pengujian *response time* dilakukan untuk mengetahui performa server dan REST API ketika diakses oleh banyak pengguna dalam waktu yang hampir bersamaan. Pengujian ini diujikan dengan jumlah pengguna yang berbeda yaitu 100, 500, 1000, 2000 dan 3000. Aplikasi Apache JMeter digunakan untuk melakukan pengujian ini.

Thread Group digunakan untuk menyimulasikan jumlah pengguna di aplikasi JMeter. Satu thread diasumsikan sebagai satu pengguna. Sehingga pengujian dengan 100 pengguna akan menggunakan 100 *thread*. Parameter Ramp-Up-Period menentukan waktu yang dibutuhkan untuk mempersiapkan *thread*. Karena pengujian ini mengukur performa dan *response time* server ketika diakses hampir bersamaan, maka nilai Ramp-Up-Period di jadikan satu detik. Parameter Loop Count akan diisi dengan nilai satu karena pengujian hanya dilakukan satu kali.

Fitur HTTP Request digunakan untuk melakukan *request* ke server. Pengujian ini akan menggunakan *endpoint* mendapatkan data destinasi wisata. Penjelasan dari *endpoint* tersebut dapat dilihat pada tabel 4.4 implementasi api endpoint halaman 110 nomor satu (1).

Untuk melihat laporan dari pengujian *JMeter*, dibutuhkan beberapa *Listener*. Pengujian ini menggunakan tiga *Listener* yaitu *Summary Report*, *View Results Tree* dan *View Results in Table*. *Summary Report* digunakan untuk melihat ringkasan dari pengujian. Dari laporan tersebut dapat dilihat rata-rata, minimum dan maksimum *response time*, *request* yang dapat ditangani oleh server per detik (*throughput*) dan persentase *error*. Informasi lebih detail mengenai hasil pengujian dapat dilihat di *View Results in Table*. Sedangkan *View Results Tree* menyajikan hasil informasi *response* server dari setiap *request* yang dilakukan. Hasil pengujian yang ditampilkan pada bab ini mengacu pada *Summary Report*.

5.3.1. Pengujian 100 Pengguna

Berdasarkan Tabel 5.17, pengujian 100 pengguna pada server menghasilkan tingkat keberhasilan 100 persen dan rata-rata *response time* 0,3 detik. *Response time* tersebut dapat dikatakan cepat dan dapat diandalkan.

Tabel 5.17 Hasil Pengujian *Response Time* 100 Pengguna

Kode Pengujian		RT001
Jumlah Pengguna		100
Persentase Keberhasilan		100%
<i>Response Time</i> (Detik)	Minimum	0,139
	Maksimum	0,641
	Rata-Rata	0,302
<i>Throughput</i>		76,8

5.3.2. Pengujian 500 Pengguna

Berdasarkan Tabel 5.18, pengujian 500 pengguna pada server menghasilkan tingkat keberhasilan 100 persen dan rata-rata *response time* 2,7 detik. *Response time* tersebut dapat dikatakan cepat dan dapat diandalkan.

Tabel 5.18 Hasil Pengujian *Response Time* 500 Pengguna

Kode Pengujian		RT002
Jumlah Pengguna		500
Persentase Keberhasilan		100%
<i>Response Time</i> (Detik)	Minimum	0,132
	Maksimum	5,150
	Rata-Rata	2,703
<i>Throughput</i>		78,9

5.3.3. Pengujian 1000 Pengguna

Berdasarkan Tabel 5.19, pengujian 1000 pengguna pada server menghasilkan tingkat keberhasilan 100 persen dan rata-rata *response time* 5,7 detik. *Response time* tersebut dapat dikatakan cepat dan dapat diandalkan.

Tabel 5.19 Hasil Pengujian *Response Time* 1000 Pengguna

Kode Pengujian		RT003
Jumlah Pengguna		1000
Persentase Keberhasilan		100%
<i>Response Time</i> (Detik)	Minimum	0,162
	Maksimum	11,318
	Rata-Rata	5,780
<i>Throughput</i>		80

5.3.4. Pengujian 2000 Pengguna

Berdasarkan Tabel 5.19, pengujian 2000 pengguna pada server menghasilkan tingkat keberhasilan 98,45 persen dan rata-rata *response time* 12 detik. Terdapat beberapa *request* yang gagal menerima *response* dari server, namun persentase keberhasilan masih diatas 95 persen, *web service* masih dapat diandalkan.

Tabel 5.20 Hasil Pengujian *Response Time* 2000 Pengguna

Kode Pengujian		RT004
Jumlah Pengguna		2000
Persentase Keberhasilan		98,45%
<i>Response Time</i> (Detik)	Minimum	0,225
	Maksimum	34,224
	Rata-Rata	12,648
<i>Throughput</i>		56,7

5.3.5. Pengujian 3000 Pengguna

Berdasarkan Tabel 5.21, pengujian 3000 pengguna pada server menghasilkan tingkat keberhasilan 97,87 persen dan rata-rata *response time* 10,5 detik. Tingkat keberhasilan 97,87 persen

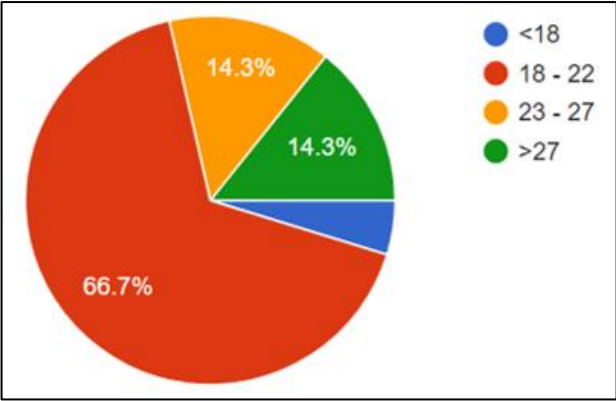
menandakan bahwa terdapat peningkatan jumlah *request* yang gagal menerima *response* dari server dibandingkan dari hasil pengujian dengan jumlah 2000 pengguna. Namun karena persentase keberhasilan yang masih diatas 95 persen, *web service* masih dapat diandalkan.

Tabel 5.21 Hasil Pengujian Response Time 3000 Pengguna

Kode Pengujian		RT005
Jumlah Pengguna		3000
Persentase Keberhasilan		97,87
Response Time (Detik)	Minimum	1,668
	Maksimum	39,573
	Rata-Rata	18,152
Throughput		71,9

5.4. Evaluasi Pengujian

Pada bab ini dijelaskan mengenai hasil pengujian fungsionalitas dan kualitas yang didapatkan dari kuesioner yang di selenggarakan pada tanggal 05 Juli 2018, pukul 11.00-selesai, dapat dilihat *chart* pada Gambar 5.17 pengujian dilakukan oleh 21 responden dari beberapa kalangan usia guna mengetahui kebutuhan pengguna dikalangan generasi millennial apakah sudah sesuai serta untuk pengembangan selanjutnya dimana rentan usianya 4,8% berusia <18 tahun, 66,7% berusia 18-22 tahun, 14,3% berusia 23-27 tahun dan 14,3% berusia >27 tahun, serta mengenai hasil pengujian *response time* yang dilakukan menggunakan aplikasi Apache.



Gambar 5.17 Rentan usia responden

5.4.1. Evaluasi Pengujian Fungsionalitas

Pada Tabel 5.22 merupakan hasil evaluasi fungsionalitas terkait keberhasilan pengujian fungsionalitas sistem. Dengan presentase keberhasilan diatas 90%.

Tabel 5.22 Hasil Evaluasi uji fungsionalitas

No	Nama	Kode Pengujian											
		SP-01	SP-02	SP-03	SP-04	SP-05	SP-06	SP-07	SP-08	SP-09	SP-10	SP-11	SP-12
1	Vania	1	0	1	1	1	1	1	1	1	1	1	
2	Afiif Naufal	1	1	1	1	1	1	1	1	1	1	1	
3	Lucha Kamala	1	1	1	1	1	1	0	1	1	1	1	
4	Dewangga Okta	1	1	1	1	1	1	1	1	1	1	1	
5	Brigita Ayu	1	1	1	1	1	1	1	0	0	1	1	
6	Rayung Laksita	1	1	1	1	1	1	1	0	0	1	1	
7	Alfitriah Nurramadhan	1	1	1	1	1	1	1	1	1	1	1	1
8	Widiyanti	1	1	1	1	1	1	1	1	1	1	1	1
9	Debora Diah Ayu	1	1	1	1	1	1	1	1	1	1	1	1
10	Patricia Bela	1	1	1	1	1	1	1	1	1	1	1	1
11	Rahma Sabila	1	1	1	1	1	1	1	1	1	1	1	1
12	Rahmad Ramadhani	1	1	1	1	1	1	1	1	1	1	1	1
13	Arruum Murti	1	1	1	1	1	1	1	1	1	1	1	1
14	Faishal Ilham	1	1	1	1	1	1	1	1	1	1	1	1
15	Magista Bela	1	1	1	1	1	1	1	1	1	1	1	1
16	Muhammad Humam	1	1	1	1	1	1	1	1	1	1	1	1
17	Muhammad Hakim	1	1	1	1	1	1	1	1	1	1	1	1
18	Kevin Rifaldi	1	1	1	1	1	1	1	1	1	1	1	1
19	Khevin Putra	1	1	1	1	1	1	1	1	1	1	1	1
20	Sekar Sari	1	1	1	1	1	1	1	1	1	1	1	1
21	Ulya Asyhari	1	1	1	1	1	1	1	1	1	1	1	1
Presentase Keberhasilan		100%	95.2%	100%	100%	100%	100%	95.2%	90.5%	90.5%	100%	100%	100%

5.4.2. Evaluasi Pengujian Response Time

Perbandingan hasil pengujian *response time* REST API dapat dilihat pada Tabel 5.23. Berdasarkan data pada tabel tersebut, tingkat keberhasilan pengujian semakin menurun ketika jumlah pengguna bertambah namun perubahannya tidak terlalu signifikan. Jumlah pengguna 100, 500 dan 1000 memiliki tingkat keberhasilan sempurna. Jumlah pengguna 2000 memiliki tingkat keberhasilan 98,45 persen. Sedangkan jumlah pengguna 3000 memiliki tingkat keberhasilan 97,87 persen. Terdapat penurunan persentase tingkat keberhasilan, namun karena masih diatas 95 persen, maka dapat dikatakan *web service* masih cepat dan dapat diandalkan.

Dilihat dari segi *response time*, jumlah pengguna 100 memiliki *response time* yang sangat cepat yaitu 0,3 detik untuk setiap *request*. Sedangkan jumlah pengguna 500 dan 1000 memiliki *response time* masing-masing 2,7 detik dan 5,7 detik. *Response time* untuk jumlah pengguna 2000 dan 3000 memiliki *response time* masing-masing 12,6 detik dan 18,1 detik.

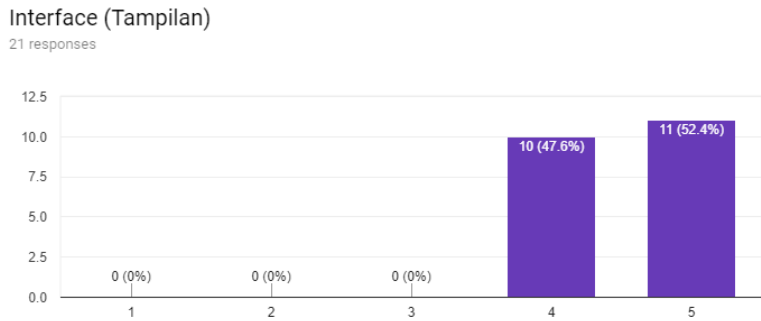
Tabel 5.23 Perbandingan Hasil Pengujian Response Time

Kode Pengujian Response Time	Jumlah Pengguna	Tingkat Keberhasilan	Rata-Rata Response Time (Detik)
RT001	100	100%	0,302
RT002	500	100%	2,703
RT003	1000	100%	5,780
RT004	2000	98,45%	12,648
RT005	3000	97,87%	18,152

5.4.3. Evaluasi Pengujian Kualitas

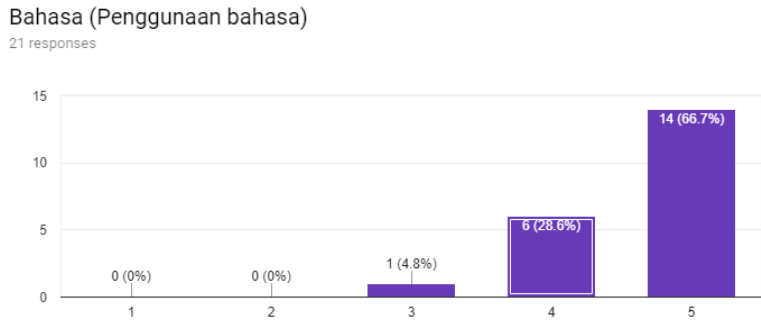
Pada Gambar 5.18 merupakan hasil penilaian uji kualitas terkait *interface* (tampilan) sistem meliputi desain, pemilihan warna, pemilihan dan peletakan *button*, peletakan menu, pemberian keterangan dan penggunaan penunjuk (*navigation*). Dengan hasil 47,6% pengguna menilai tampilan aplikasi sudah

baik dan 52,4% pengguna lainnya menilai tampilan aplikasi sudah sangat baik.



Gambar 5.18 Hasil evaluasi uji kualitas tampilan sistem

Pada Gambar 5.19 merupakan hasil penilaian uji kualitas terkait Bahasa, penggunaan bahasa, pemilihan bahasa dan istilah pada sistem. Dengan hasil 4,8% pengguna menilai penggunaan bahasa cukup baik, 28,6% pengguna menilai penggunaan bahasa sudah baik dan 66,7% pengguna menilai penggunaan bahasa sudah sangat baik.



Gambar 5.19 Hasil evaluasi uji kualitas penggunaan bahasa pada sistem

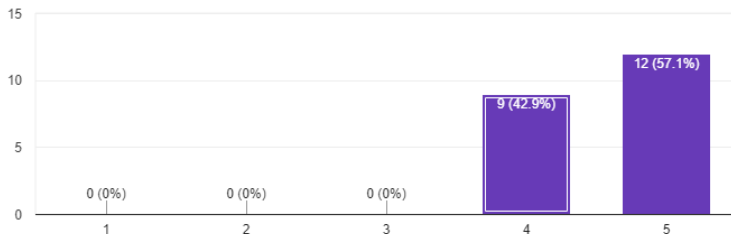
Pada Gambar 5.20 merupakan hasil penilaian uji kualitas terkait kemudahan penggunaan dimana sistem harus memenuhi kriteria:

1. Mudah dipahami baik itu konsep, penggunaan istilah serta memahami kebiasaan pengguna.
2. Konsisten atau tidak membingungkan.
3. *Minimal surprise* atau tidak membuat kaget pengguna, dimana operasi bisa diduga prosesnya berdasar perintah yang disediakan.
4. *User guidance* adanya *caption* pada *icon* khusus.

Dengan hasil 42,9% pengguna menilai kemudahan penggunaan sudah baik dan 57,1% pengguna lainnya menilai sudah sangat baik.

Kemudahan Penggunaan

21 responses



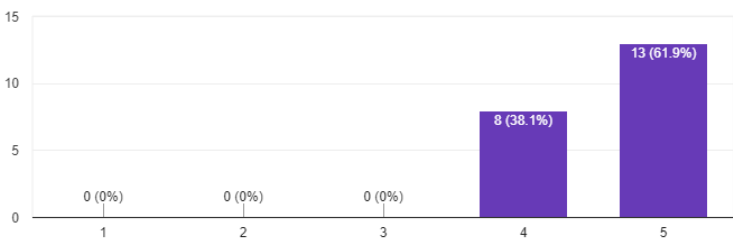
Gambar 5.20 Hasil evaluasi uji kualitas kemudahan penggunaan sistem

Berdasarkan hasil pengujian diatas, hampir seluruh skenario pengujian fungsionalitas berhasil begitupun untuk pengujian kualitas, pengguna berpendapat bawasannya sistem sudah baik pada sisi kualitasnya.

Adapun untuk kesimpulan yang diberikan para responden dapat dilihat pada Gambar 5.21 dimana 38,1% responden merasa puas dan 61,9% lainnya merasa sangat puas dengan aplikasi Magelang Tour ini.

Bagaimana penilaian anda terhadap aplikasi Magelang Tour ini?

21 responses



Gambar 5.21 Tingkat kepuasan responden

Selain itu responden juga memberikan kritik dan saran terkait sistem, dimana kritik dan saran dapat dilihat dar sebagai berikut:

Tabel 5.24 Kritik dan saran responden

Nama	Kritik dan Saran
Rayung Laksita	Lebih baik menggunakan 2 bahasa, tampilan warna lebih menarik, menampilkan lebih banyak tempa wisata. aplikasi cukup gampang dimengeri.
Vania	Bagus
Lucha Kamala	Tampilan warnanya lucu, saya suka. saran tampilan halaman untuk rekomendasi diperbaiki lagi agar terlihat lebih bagus
Dewangga Okta	Tadi sempat gagal karena location nya gak dinyalakan tapi overall udah bagus
Brigita Ayu	Tampilannya terlalu sederhana, warnanya pink monoton, waktu mau registrasi bingung mau klik mana, sebaiknya diberi perbedaan warna atau jenis font. fitur sudah lengkap dan sangat membantu.
Alfitrah N	Tampilannya girly banget ya
Widiyanti	Fiturnya banyak banget, jadi agak bingung
Debora Diah Ayu	Udah sih puas, fiturnya lengkap banget kalo dibanding sama aplikasi lain yang uda ada, tampilannya juga unyu banget
Patricia Bela	Kurang begitu paham bahasa inggris, jadi agak bingung

Rahmad Ramadhani	Perlu dicoba nih pas aku kemagelang, mungkin bisa dikembangkan untuk pilihan kota lain yang potensi wisatanya tinggi juga
Arruum Murti Ayu	Aku suka warnanya! gemes parah. oh iya saran, mungkin bisa ditambahin fitur semacam rute untuk penggunaan transportasi umum gitu
Faishal I	Udah bagus kok
Magista Bela	Keren keren, mungkin bisa ditambahin fitur penginapan terdekat dari tempat wisatany.
Kevin Rifaldi	Saran, diawal dikasih option pemakaian bahasa, biar orang awam juga bisa pake hehe
Sekar Sari	Suka sama aplikasinya, tampilan simple pilihan warnanya cute, fitur lengkap dan sesuai kebutuhan sama anak-anak jaman sekarang
Afiif N	Sudah baik, sangat membantu, mudah dioperasikan
Rahma Sabila	Ngebantu banget sumpah, meskipun fiturnya banyak tapi desainnya simple jadi gampang dioperasiin
Khevin P	No comment. udah bagus banget
M Humam	Tampilannya simple, gampang di operasiin
M Hakim	Bagus. ngebantu. pemilihan warnanya boleh juga, nyala
Ulya Asyhari	Fiturnya banyak sekali, mungkin kalau mom kaya saya gini cuma yang dipake fitur dasarnya aja, tapi overall bagus apalagi untuk anak jaman sekarang, ya mungkin bedanya yang pengguna tua cuma pake fitur dasarnya aja untuk cari info wisata tapi kalau yang anak muda make semua fiturnya

Dari pengujian kualitas dapat disimpulkan, bawasannya pengguna dengan rentan usia <18 tahun hingga 22 tahun yang merupakan kawula muda yang mana sangat menyukai fitur dari aplikasi dan mengoptimalkan seluruh fitur yang ada, selanjutnya untuk pengguna dengan rentan usia 23 tahun hingga 27 tahun juga menggunakan seluruh fitur yang ada, hanya saja tidak semua pengguna menggunakan fitur untuk *share* foto, dan yang terakhir untuk pengguna dengan rentan usia diatas 27 tahun lebih mengutamakan atau hanya menggunakan fitur utama dari aplikasi yaitu untuk mengetahui informasi wisata saja.

BAB VI

PENUTUP

Pada bab ini akan dibahas kesimpulan yang diperoleh berdasarkan hasil uji coba aplikasi untuk menjawab rumusan masalah dan saran mengenai pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

6.1. Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, maka dapat disimpulkan sebagai berikut.

1. Pengambilan data wisata dari basis data menggunakan konsep REST telah berhasil dilakukan. REST API untuk aplikasi Magelang Tour berhasil diimplementasikan menggunakan bahasa pemrograman php dan berdasarkan hasil pengujian *response time*, REST API memiliki tingkat keberhasilan diatas 95%. Sehingga bisa ditarik kesimpulan bahwa REST API mampu diandalkan untuk menangani pengguna hingga jumlah kurang lebih 3000 pengguna.
2. Mekanisme pengawasan administrator terhadap komentar negatif dilakukan dengan dua cara, yang pertama filter berdasar bank data dan secara manual, mengecek satu per satu hasil *review* yang masuk ke halaman website administrator.
3. Rancangan dan implementasi aplikasi panduan wisata dengan fitur filter kategori, melihat informasi destinasi wisata yang lengkap dan *up to date*, fitur rekomendasi wisata maupun *update* informasi wisata serta fitur galeri pengguna, dirasa sudah sesuai dengan kebutuhan pengguna era millennial berdasar hasil pengujian kebutuhan fungsional & kualitas serta hasil kepuasan pengguna.

6.2. Saran

Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan. Berikut

merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang.

1. Aplikasi ini dapat dikelola oleh Dinas Pariwisata setempat.
2. Mekanisme pengumpulan data perlu dikembangkan lebih lanjut agar dapat dilakukan secara otomatis.
3. Antarmuka aplikasi masih sederhana sehingga diperlukan perkembangan lebih lanjut.

DAFTAR PUSTAKA

- [1] I. Tourism, "About Wonderful Indonesia," Indonesia Tourism, 2017. [Online]. Available: <http://indonesia-tourism.com/>. [Accessed 28 December 2017].
- [2] P. Rumah, "Siapa itu Generasi Millennial?," Rumah Millennials, 8 March 2017. [Online]. Available: <http://rumahmillennials.com/siapa-itu-generasi-millennials/#.W0iGJtIzbIU>. [Accessed 2 July 2018].
- [3] M. D. Gayati, "Destinasi digital potensi tarik wisatawan milenial," Antar News, 5 April 2018. [Online]. Available: <https://www.antaraneews.com/berita/698964/menpar-destinasi-digital-potensi-tarik-wisatawan-milenial>. [Accessed 2 July 2018].
- [4] M. Pertiwi, "Generasi Milenial, Pasar Potensial Industri Pariwisata," Kompas, 14 September 2016. [Online]. Available: <https://travel.kompas.com/read/2016/09/14/110300127/generasi.milenial.pasar.potensial.industri.pariwisata>. [Accessed 02 July 2018].
- [5] "Android_(operating_system)," Android, [Online]. Available: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)). [Accessed 2 January 2018].
- [6] "Android_fundamental," Google, [Online]. Available: <https://developer.android.com/guide/components/fundamentals.html>. [Accessed 12 January 2018].
- [7] "LocationListener," Google, [Online]. Available: <https://developer.android.com/reference/android/location/LocationListener.html>. [Accessed 20 December 2017].
- [8] G. Osahon, "Consuming REST API using Retrofit Library in Android," AndroidPub, 19 March 2017. [Online]. Available: <https://android.jlelse.eu/consuming-rest-api-using-retrofit-library-in-android-ed47aef01ecb>. [Accessed 02 July 2018].
- [9] Feridi, "Mengenal RESTful Web Services," CodePolitan, 25 February 2016. [Online]. Available: <https://www.codepolitan.com/mengenal-restful-web-services>. [Accessed 28 December 2017].

- [10] W3C®, "Web Services Architecture," W3C®, 11 February 2004. [Online]. Available: <https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>. [Accessed 8 June 2017].
- [11] "Pengenalan JSON," [Online]. Available: <http://www.json.org/json-id.html>. [Accessed 28 December 2017].
- [12] Microsoft, "Microsoft Releases SQL Server 2012 to Help Customers Manage "Any Data, Any Size, Anywhere"," MICROsoft News Center, 7 March 2012. [Online]. Available: <http://www.microsoft.com/Presspass/press/2012/mar12/03-06SQLServer12PR.mspx>. [Accessed 5 January 2018].
- [13] Microsoft, "SQL Server Documentation," Microsoft, 10 October 2017. [Online]. Available: <https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation>. [Accessed 5 January 2018].
- [14] A. El-Rabanny, "Introduction to GPS The Global Positioning System," Boston, Artech House, 2002.
- [15] "PHP," PHP, [Online]. Available: <https://en.wikipedia.org/wiki/PHP>. [Accessed 01 January 2018].
- [16] "Google Maps API," Google, [Online]. Available: <https://developers.google.com/maps/web-services/overview#WebServices>. [Accessed 30 December 2017].
- [17] Google, "Introduction to the Google Maps Android API," Google, 17 June 2017. [Online]. Available: <https://developers.google.com/maps/documentation/android-api/intro>. [Accessed 28 December 2017].
- [18] "Inilah Kriteria Objek Wisata Idaman Wisatawan di Era Kekinian," 9 December 2019. [Online]. Available: <http://www.wisatajabar.com/2017/12/inilah-kriteria-objek-wisata-idaman.html>. [Accessed 31 January 2018].

BIODATA PENULIS



Penulis, **Nurul Azizah**, lahir di Magelang, 14 Maret 1996. Penulis menempuh pendidikan sekolah dasar di SD Negeri Kalinegoro 5, Magelang. Melanjutkan pendidikan sekolah menengah pertama di SMP Negeri 4 Magelang dan selanjutnya di SMA Negeri 3 Magelang. Selanjutnya penulis melanjutkan pendidikan sarjana di Jurusan Teknik Informatika, Fakultas Teknologi dan Informasi, Institut Teknologi Sepuluh Nopember Surabaya. Selama kuliah, penulis aktif menjadi pengurus UKM Unit Kegiatan Tari dan Karawitan divisi modern dance. Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat penulis mengambil bidang minat Rekayasa Perangkat Lunak (RPL). Penulis dapat dihubungi melalui instagram dengan nama pengguna @uyunna atau melalui surel na.azizah14@gmail.com.